

## REPORT DOCUMENTATION PAGE

Form Approved  
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimates or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 19th June 96	3. REPORT TYPE AND DATES COVERED Final Progress Rept, July, 95 - June 96	
4. TITLE AND SUBTITLE Optical Pulse Coding for Maximum Data Rate		5. FUNDING NUMBERS DAAH04-95-C-0051	
6. AUTHOR(S) Terence W. Barrett, Ph.D.			
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES) BSEI 1453 Beulah Road Vienna, VA 22182		8. PERFORMING ORGANIZATION REPORT NUMBER 12	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211		10. SPONSORING / MONITORING AGENCY REPORT NUMBER ARC 34591.1-MA-SDI	
11. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.		12 b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  This study shows the benefit of a <i>two-stage hierarchy</i> of codes for maximum data rate transmission. At the first stage, <i>orthogonal codes</i> define the microchannel or user, identified by a first (temporal) matched filter. Once a code is identified by this filter, a second (temporal) matched filter identifies the <i>pulse-position-modulated</i> data with error correction, hence <i>error-correcting codes</i> are required for the second stage. Congruence codes are recommended for orthogonal codes (hyperbolic, quadratic, cubic and quartic) and Lexicographic codes for error correction codes. Besides a <i>hierarchical backbone topology</i> , a <i>multidimensional coding scheme</i> intermixing CMDA, TDMA and WDM provides signal spreading techniques, but they are <i>spread time techniques</i> , as opposed to spread spectrum techniques. A major part of this study addressed the impact of <i>symmetry principles</i> on code design to achieve optimum properties. This study provides many examples of heretofore unknown symmetries underlying code design. Future work will address the use of symmetry in designing optimum codes.			
4. SUBJECT TERMS Optical Communications, Optical orthogonal codes; Greedy algorithm, Lexicographic codes		15. NUMBER OF PAGES 132	
		16. PRICE CODE	
7. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

# DISCLAIMER NOTICE



THIS DOCUMENT IS BEST  
QUALITY AVAILABLE. THE  
COPY FURNISHED TO DTIC  
CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO  
NOT REPRODUCE LEGIBLY.

ARO Contract DAAH04-95-C-0051  
 Effective Date 1 July 1995  
 Requisition: P-34591-MA-SDI  
 TITLE: OPTICAL PULSE CODING FOR MAXIMUM DATA RATE

US Army Research Office  
 ATTN: AMXRO-ICA (Ramseur)  
 P.O. Box 1211 Research Triangle Park, NC 27709-2211

COR: US Army Research Office  
 ATTN: Dr. Jagdish Chandra  
 Mathematical and Computer Sciences Division  
 P.O. Box 12211  
 Research Triangle Park, NC 27709-2211

### FINAL TECHNICAL REPORT

Terence W. Barrett, Ph.D., BSEI, 1453 Beulah Road, Vienna, VA 22182  
 Tel: 703-759-4518; E-Mail: barrett506@aol.com

June 19th, 1996

*Summary:*

The major findings and recommendations of this study are:

- The work over the past 12 months has shown the benefit of a *two-stage hierarchy* of codes. At the first stage, *orthogonal codes* define the microchannel or user, identified by a first (temporal) matched filter. Once a code is identified by this filter, a second (temporal) *error-correcting codes* are required for the second stage. The recommendations of this study are that congruence codes be used as orthogonal codes (hyperbolic, quadratic, cubic and quartic) and Lexicographic codes for error correction codes.
- Besides a *hierarchical backbone topology*, a *multidimensional coding scheme* intermixing CMDA, TDMA and WDM provides the possibility of the highest data rates.
- These recommended techniques provide signal spreading techniques, but they are *spread time techniques*, as opposed to spread spectrum techniques.
- A major part of this study addressed the impact of *symmetry principles* on code design to achieve optimum properties. This study provides many examples of heretofore unknown symmetries underlying code design. Future work will address the use of symmetry in designing optimum codes.

*Contents:*

- 1.0 Results of Contract DAAH04-95-C-0051
  - 1.1 Introduction and General Discussion
  - 1.2 Code Bounds
  - 1.3 Bit Error Bounds
  - 1.4 Code Design
    - 1.4.1 Perfect Error-Correcting Codes
    - 1.4.2 The Trade-off between Low Out-of-Phase Auto-Correlation and Low Cross-Correlation in Orthogonal Codes.
  - 1.5 Symmetry in Code Design
- 2.0 Hardware Analysis
- 3.0 Recommendations
- 4.0 Future Work

## 1.0 Results of Contract DAAH04-95-C-0051

### 1.1 Introduction and General Discussion

The discussion of the results to follow is facilitated by use of the following nomenclature. Referring to Fig. 1, orthogonal codes are defined over a superframe of slots. A single pulse occurs only once per frame and the complete pulse code sequence is transmitted over the total time of the superframe of frames.

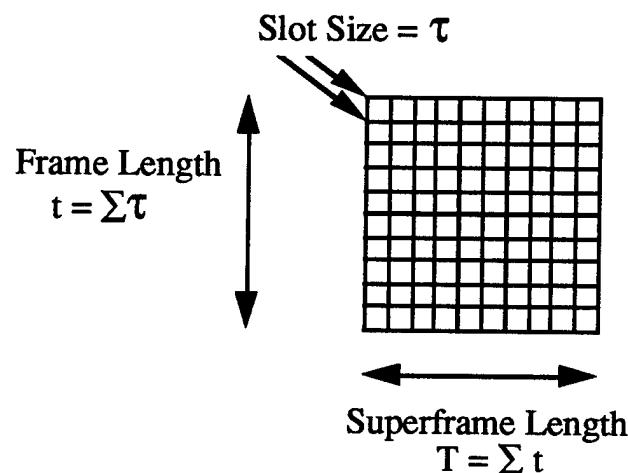


Fig. 1. Orthogonal codes are defined over a superframe of frames. A pulse of each orthogonal code occurs only once per frame and the total code sequence of pulses occurs completely over the superframe time. In this example, the frame size is 10 slots and there are 10 frames, so the size of the superframe is 100 slots. Therefore a complete orthogonal code is 10 pulses spread over the superframe time.

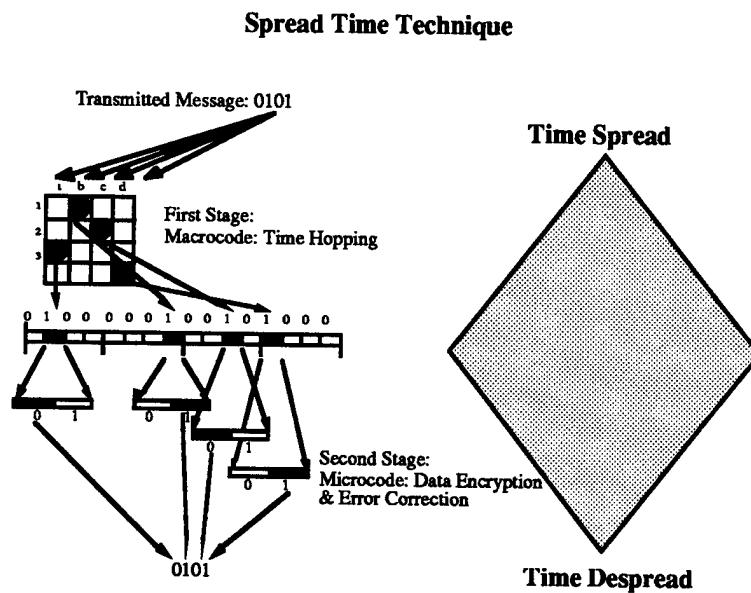


Fig. 2. Graphical representation of the Spread Time Technique with coding at a first stage identifying the channel and coding at a second stage by pulse-position modulation providing data encryption and error-correction.

19960911 050

Two major types of codes will be discussed below: (1) orthogonal codes; and (2) error-correcting codes. The orthogonal codes are generated over a matrix field and are transferred to a time hopping representation by the method shown in Fig. 2, where a number of frames of a superframe are shown. The *spread time technique* reduces noise during the transmission phase (as does the spread spectrum technique).

The work over the past 12 months has shown the benefit of a *two-stage hierarchy* of codes. The *orthogonal codes* define the microchannel or user, and, once a code is identified by a first matched filter, a second matched filter identifies the pulse-position-modulated data with error correction, hence *error-correcting codes* are required for the second stage. The two stages are represented in Fig. 3.

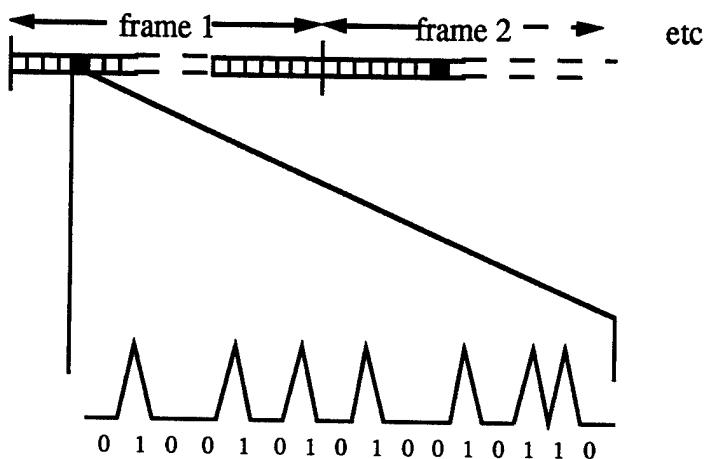


Fig. 3. Two-stage encoding. The user, or channel is defined by the orthogonal code. A second matched filter, positioned behind the first, decodes data and applies error-correction. Hence error-correcting codes are required for the second stage. The word size of the second stage is dependent upon the availability of optical devices capable of providing pulse modulations representing  $n$ -bit words. Here, a 16-bit word is shown.

Using the two-stage principle of encoding, multidimensional coding, and presuming device capabilities which are either available now in the laboratory, or might be in the future (see Section 2), predictions can be made concerning the data rate achievable for high data rate channels. The algorithm of Table 1 provides those predictions.

This algorithm takes into consideration that with present technology at least 4 wavelengths are available now in an optical fiber channel. Therefore there is the possibility of either using this wavelength diversity to define separate channels, or of using the diversity for frequency hopping with time hopping. As there is also the possibility of wavelet generation with diversity in both scale and orthogonality, the channel capacity can be increased yet further. However, it must be recognized that there are trade-offs between frequency and wavelet diversity due to device limitations (addressed in Section 2.0).

**Table 1: An Example Algorithm - Channel Data Rate Estimate with Two-Level Coding & Device and Input-Output Capability Assumed§**

		2-bit word	4-bit word	8-bit word	16-bit word	32-bit word†
<b>One Code (MicroChannel)</b>						
Min. MicroPulse Durat. Assumed	100 femtosec.					
MacroPulse & MacroSlot Duration	1 picosecond					
MacroSlot Repetition Frequency	1 TeraHertz					
<b>One Code/MicroChannel Data Rate</b>						
	2 Gigabits/s.	4 Gigabits/s.	8 Gigabits/s.	16 Gigabits/s.	32 Gigabits/s.	
<b>MacroChannel</b>						
For a 1021 x 1021 length code:						
MacroSlot Rate	1 TeraHertz					
Frame Rate	1 GigaHertz					
SuperFrame Rate	1 MegaHertz					
MicroSlot Duration	500 femtos.	250 femtos.	125 femtos.	62.5 femtos.	31.25 femtos.	15.625 femtos.
Timing Accuracy Required	±250 femtos.	±125 femtos.	±62.5 femtos.	±31.25 femtos.	±15.625 femtos.	
<b>One Code/MicroChannel Data Rate</b>						
1021 Code/MacroChannel	2 Gigabits/s.	4 Gigabits/s.	8 Gigabits/s.	16 Gigabits/s.	32 Gigabits/s.	
2 Terabits/s.	4 Terabits/s.	8 Terabits/s.	16 Terabits/s.	32 Terabits/s.	64 Terabits/s.	
<b>Wavelengths Available (Now)</b>	4	4 Terabits/sec.	16 Terabits/sec.	32 Terabits/sec.	64 Terabits/sec.	128 Terabits/sec.
<b>Wavelengths Available (Future)</b>	8	8 Terabits/sec.	32 Terabits/sec.	64 Terabits/sec.	128 Terabits/sec.	256 Terabits/sec.

§ A minimum resolvable micropulse duration of 100 femtoseconds is assumed.

† Shaded columns assume a minimum resolvable micropulse duration of less than 100 femtoseconds.

The major results of the 12-month study, as well as the proposed future work, are (Table 2):

Table 2: Major Results and Discoveries of the 12-month Study			
Major Results & Discoveries	Tables & Figures	Proposed Future Work	Reports
1. Increased data rate using the availability of frequency and wavelet diversity. 2. The number of users possible defined by orthogonal codes without interference.	Fig.s 4-6  Fig. 7	The present results are generic and need to be interpreted within the context of actual device capability. Presently only 4 wavelengths are available, and the diversity of wavelet in terms of scale and orthogonality has not been determined.	August, 1975, Report
1. Any orthogonal code can be identified by only two inputs to a code generator and the third predicted. That is, these orthogonal code sequences are nonMarkovian and can be generated by programmable logic arrays. 2. A hidden symmetry was discovered underlying orthogonal codes in terms of the polynomial associated with each hopping sequence 3. A hidden symmetry was discovered underlying the orthogonal codes in terms of an increasing or decreasing generator polynomial index.	Fig. s 8-12, Fig. 15	These discoveries need find use in code design, i.e., design on the basis of the symmetry.	September, 1975, Report
1. Orthogonal code symmetries were discovered based on residue-nonresidue matrix formulations of such codes	Fig.s 13-15.	The discoveries need to find use in code design, i.e., design on the basis of the symmetry forms.	October, 1975, Report
1. Cross correlation bounds were found for time-hopping and frequency-hopping codes		Cross-correlation bounds need to be tied to code design in a predictive way. The bounds need to be derived.	November, 1975, Report
1. A back-track mathematical algorithm for time hop code generation from a given hit array was developed. Tradeoff between low out-of-phase correlation and low cross-correlation was derived. 2. Cubic congruence and quartic congruence codes generated and generator polynomial index matrix representation demonstrated. 3. A code comparison of auto- and cross-correlation properties of congruence codes completed	Fig.s 8, 13, 14.  Table 3	Back-track algorithm needs development to be a predictive tool for code properties.  Matrix representation needs development as a predictive tool.  Properties need to be linked to code design algorithm.	December, 1975, Report
1. Probability of error analysis indicates that error decreases with (1) an increase of the matched filter threshold, and (2) an increase in code length.	Fig.s 16-26	Receiver threshold properties need to be analyzed together with code properties.	January, 1996, Report

2. Algorithm developed for Lempel-Costas arrays.			
1. $\lambda=2$ optical orthogonal codes may be as good as $\lambda=1$ codes with receiver hard-limiting. 2. More pulses in a sequence does not result in a reduction in performance if one can raise the receiver threshold. 3. Surreal number representation reveals further underlying code symmetries.	Fig.s 27-36	The tradeoffs between the properties of optical orthogonal codes, without hard limiting, and with hard limiting, need to be investigated. The surreal number representation needs to be related to optimum code design.	February, 1996, Report
1. A Greedy Code Algorithm was developed that permits the study of Lexicographic-Greedy codes necessary for the second stage of the system design. 2. (5,2) and (7,4) Lexicographic-Greedy codes have been generated. 3. The standard arrays for Lexicographic-Greedy codes have revealed design details. 4. The $g$ -values of the codes have revealed underlying symmetries.	Tables 7-9 Tables 4-6 Fig.s 39-42	The Lexicographic-Greedy code algorithm offers opportunities for developing optimum error-correcting codes. The design of such codes needs to be taken further.  The discovered $g$ -value symmetries should be used in code design.	March, 1996, Report
1. Standard arrays of Lexicographic-Greedy error-correcting codes are ordered in cosets defined by the cosets' $g$ -values. The coset leaders may, or may not, be the basis of vectors defining the code.	Tables 7-9	Investigate the exact role of the coset leaders in defining a code's properties.	April, 1996, Report
1. For the first time, symmetries are shown underlying the lexicographic ordering of the $g$ -values of the Lexicographic-Greedy codes.	Fig. 43 A-D	Investigate the role of these symmetries in defining a code's properties.	April, 1996, Report
1. The Greedy code algorithm, which generates codes according to a designed distance, $d$ , at set $n$ , provides superior codes than those generated by the conventional methods of commencing with a designed $k$ , at set $n$ .	Fig.s 44-52	Develop the Greedy code algorithm for longer perfect code generation. Determine the role of the Greedy algorithm in the generation of perfect codes.	May, 1996, Report
1. The symmetries underlying the Lexicographic codes are shown. However, the most symmetric code studied is not the optimum, as the code's code-rate, $k/n$ , exceeds the channel capacity.	Fig.s 53-58	The code bounds derived are for large $n$ . Therefore, further investigation may reveal that symmetry does play a role in choosing the optimum Lexicographic code.	May, 1996, Report

A Gaussian approximation can be used to evaluate the probability of error for time hopping codes<sup>1</sup>. A derivation in Kwong et al<sup>2</sup> gives the following approximation (valid for large values of  $N$ ) for the probability of error,  $P_E$ :

<sup>1</sup>Prucnal, P.R., Santoro, M.A. & Fan, T.R., Spread spectrum fiber-optic local area network using optical processing. *J. Lightwave Technology*, LT-4, 547-554, 1986;  
Kwong, W.C., Perrier, P.A. & Prucnal, P.R., Performance comparison of asynchronous and synchronous code-division multiple-access techniques for fiber-optic local area networks. *IEEE Trans. Comm.*, 11, 1625-1634, 1991;

$$P_E = \phi\left(\frac{-P}{\sqrt{N-1}}\right),$$

where

$$\phi = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp[-\xi^2/2] d\xi$$

is the unit normal cumulative distribution function and  $P$  is a prime number, as well as the number of codes generated and equal to  $\omega$ , the number of binary ones per code sequence. Accordingly, Fig. 4 shows the (log) probability of error versus the number of simultaneous users as a function of  $P$ .

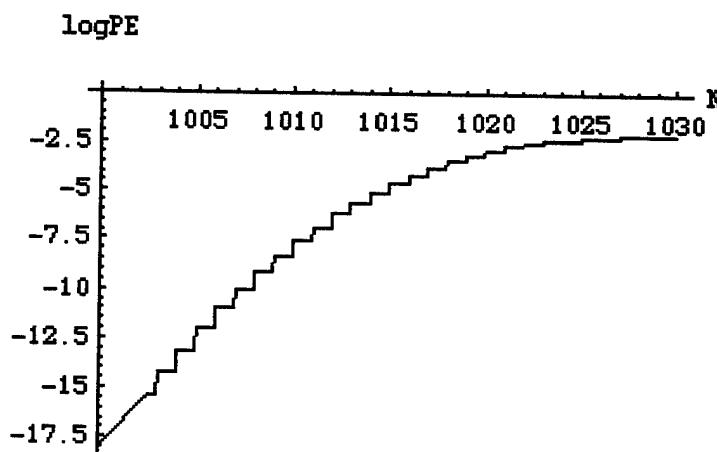


Fig. 4 Log Probability of error versus the number of simultaneous users for prime number and number of codes = 1021. This shows that pulse position modulation (PPM) CDMA can accommodate 1021 subscribers and 1008 simultaneous users with a probability of error less than  $10^{-9}$ .

Fig. 4 indicates that time hopping pulse position modulation schemes using orthogonal codes can accommodate a large number of simultaneous users at excellent error probability. Probability of Error is examined in more detail in Section 1.3.

Table 3 summarizes the results shown in Fig.s 8-12. Based on auto- and cross-correlation hit criteria and  $11 \times 11$  or  $10 \times 10$  matrices, the quadratic and quartic congruence codes are superior, followed closely by the hyperbolic, then cubic congruence codes. The Welch-Costas arrays are definitely inferior in cross-correlation properties (although superior in auto-correlation properties). The trade-off between low out-of-phase auto-correlation and low cross-correlation in orthogonal codes is examined in detail in Section 1.4.2.

---

Gagliardi, R.M., Mendez, A.J., Dale, M.R. & Park, E., Fiber-optic digital video multiplexing using optical CDMA. *J. Lightwave Technology* 11, 20-26, 1993;  
 Gagliardi, R.M., Pulse-coded multiple access in space optical communications. *IEEE J. Selected Areas in Communications*, 13, 603-608, 1995.

<sup>2</sup>Kwong, W.C., Perrier, P.A. & Prucnal, P.R., Performance comparison of asynchronous and synchronous code-division multiple-access techniques for fiber-optic local area networks. *IEEE Trans. Comm.*, 11, 1625-1634, 1991.

**Table 3** Code Comparison of Auto- and Cross-Correlation Properties

Code	Maximum Number of Hits in the Auto-Correlation Subbands	Maximum Number of Hits Across All Cross-Correlation Bands
Hyperbolic Congruence	4	3
Quadratic Congruence	2	4
Welch-Costas	2	9
Cubic Congruence	4	4
Quartic Congruence	3	3

Hyperbolic codes are shown in multidimensional form in Fig.s 5-7. The two axes, other than the time axis, represent frequency diversity and wavelet diversity. Regarding frequency diversity: as stated in Table 1 above, only 4 frequencies are now available in optical fiber transmissions. Therefore, the frequency diversity shown in these Fig.s is not now available. The same can be said for wavelet diversity. In order to apply the fine structure required in wavelet representation, the maximum modulation capacity is used, leaving no possibility of anything other than a 1 bit wave packet or transmission word. Therefore, and referring to Table 1 again, if channels are defined by wavelets, then each wave packet transmitted is no more than a 1 bit transmission.

The first section following (1.2) examines the bounds on codes imposed by Shannon's laws of transmission with minimum error.

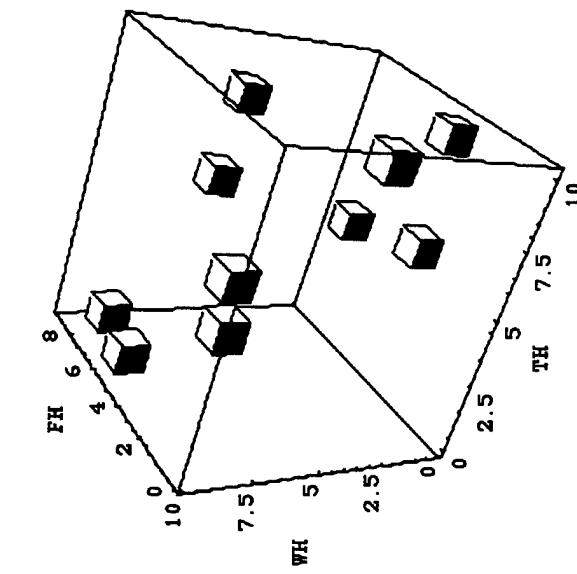


Fig. 5

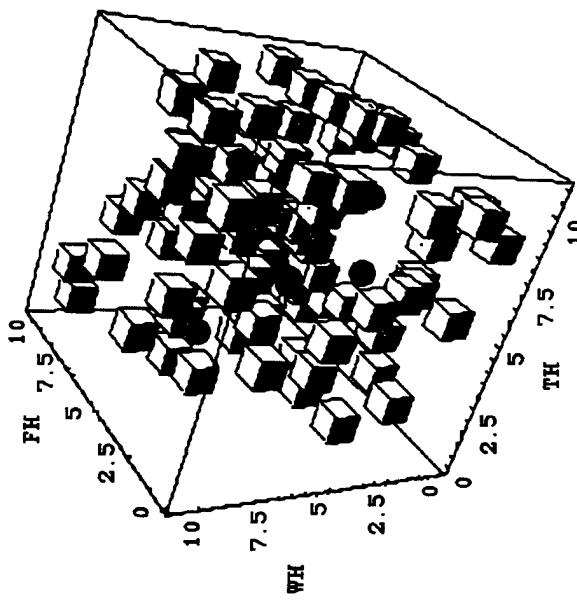


Fig. 6

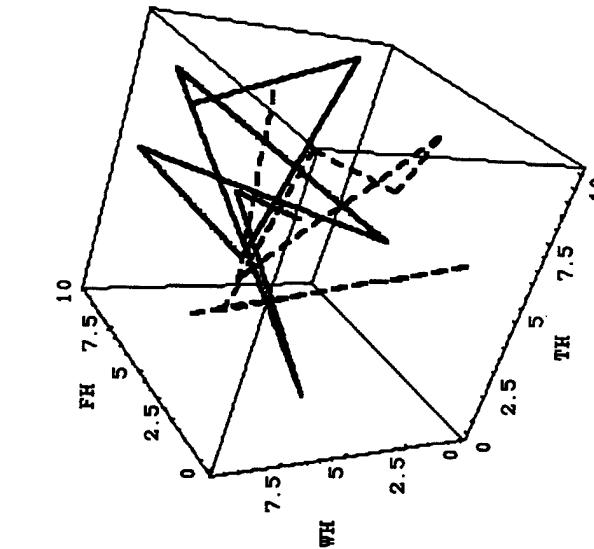


Fig. 7

Fig. 5 Hyperbolic congruence code<sup>3</sup> for  $i = 1, \text{mod } 11$ , for time hopping and with random coding for frequency and wavelet hopping. The placement operator is:  $y_i(k) = \frac{i}{k} + m \pmod{p}$ ,  $1 \leq k \leq p-1$ ,  $i = 1, 2, \dots, Q$ ,  $m \in M$ , where  $\frac{i}{k}$  is defined as the multiplicative inverse of  $k$  in the field  $J_p$ . Note that the TH slots are shown only sequentially, i.e., the TH frames have been collapsed to the single slot at which the code pulse will occur.

Fig. 6 Hyperbolic congruence codes,  $i = 1 \dots 10, \text{mod } 11$  for time hopping. The  $i = 1, \text{mod } 11$ , code is represented with spheres. Frequency hopping and wavelet hopping by random coding.

Fig. 7 Hyperbolic congruence codes for  $i = 1$  (joined by solid line),  $\text{mod } 11$ , for time hopping. Frequency and wavelet hopping by random coding.

<sup>3</sup> Maric, S.V. & Titlebaum, E.L., A class of frequency hop codes with nearly ideal characteristics for use in multiple-access spread spectrum communications and radar and sonar systems. *IEEE Trans. Comm.*, 40, 1442-1447, 1992.

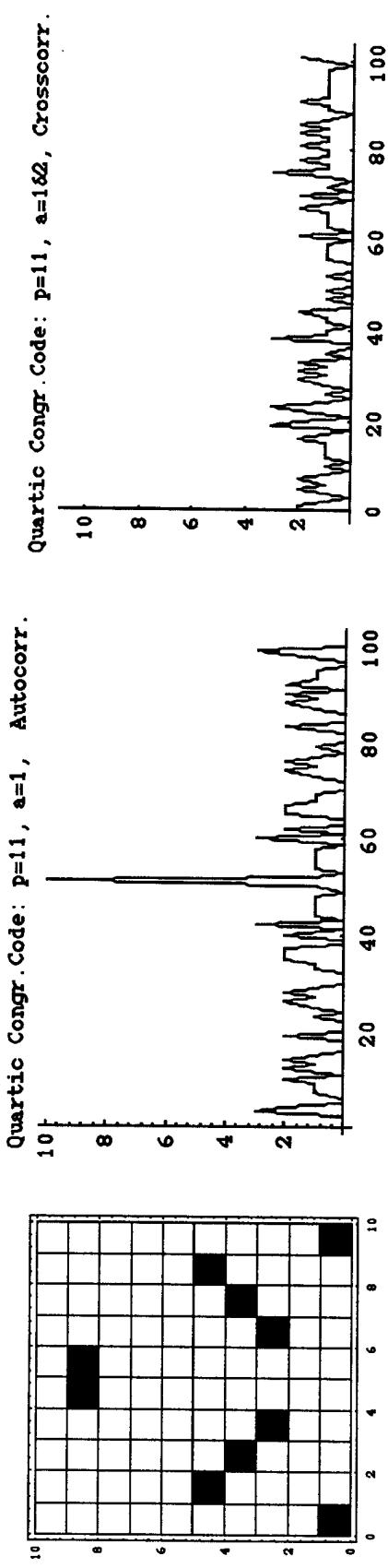
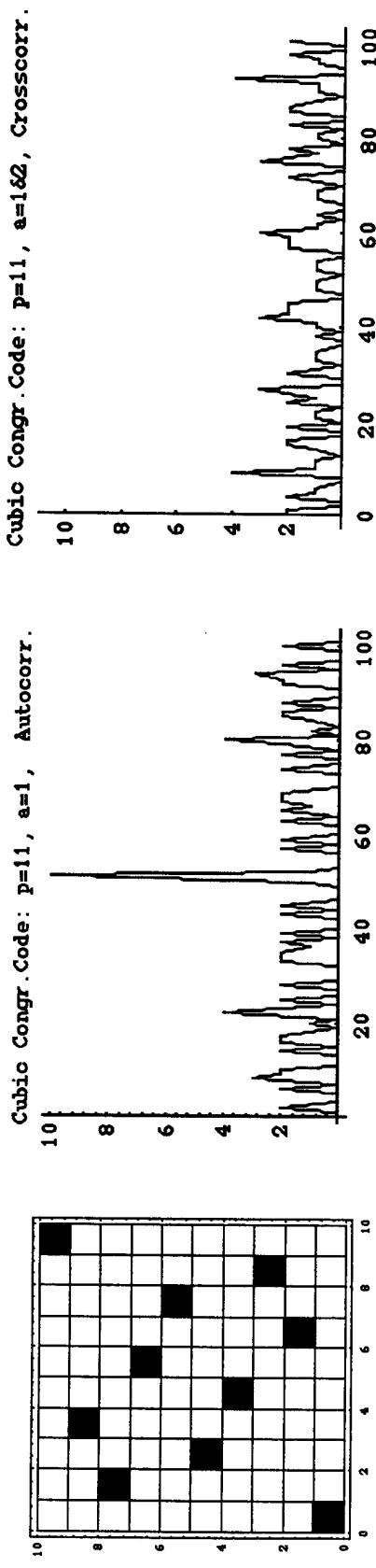


Fig. 8 Auto- and Cross-correlations of cubic<sup>4</sup> and quartic congruence codes.

---

<sup>4</sup> Maric, S.V. & Titlebaum, E.L., Frequency hop multiple access codes based upon the theory of cubic congruences. *IEEE Trans. Aerospace and Electronic Systems*, 26, 1035-1039, 1990.

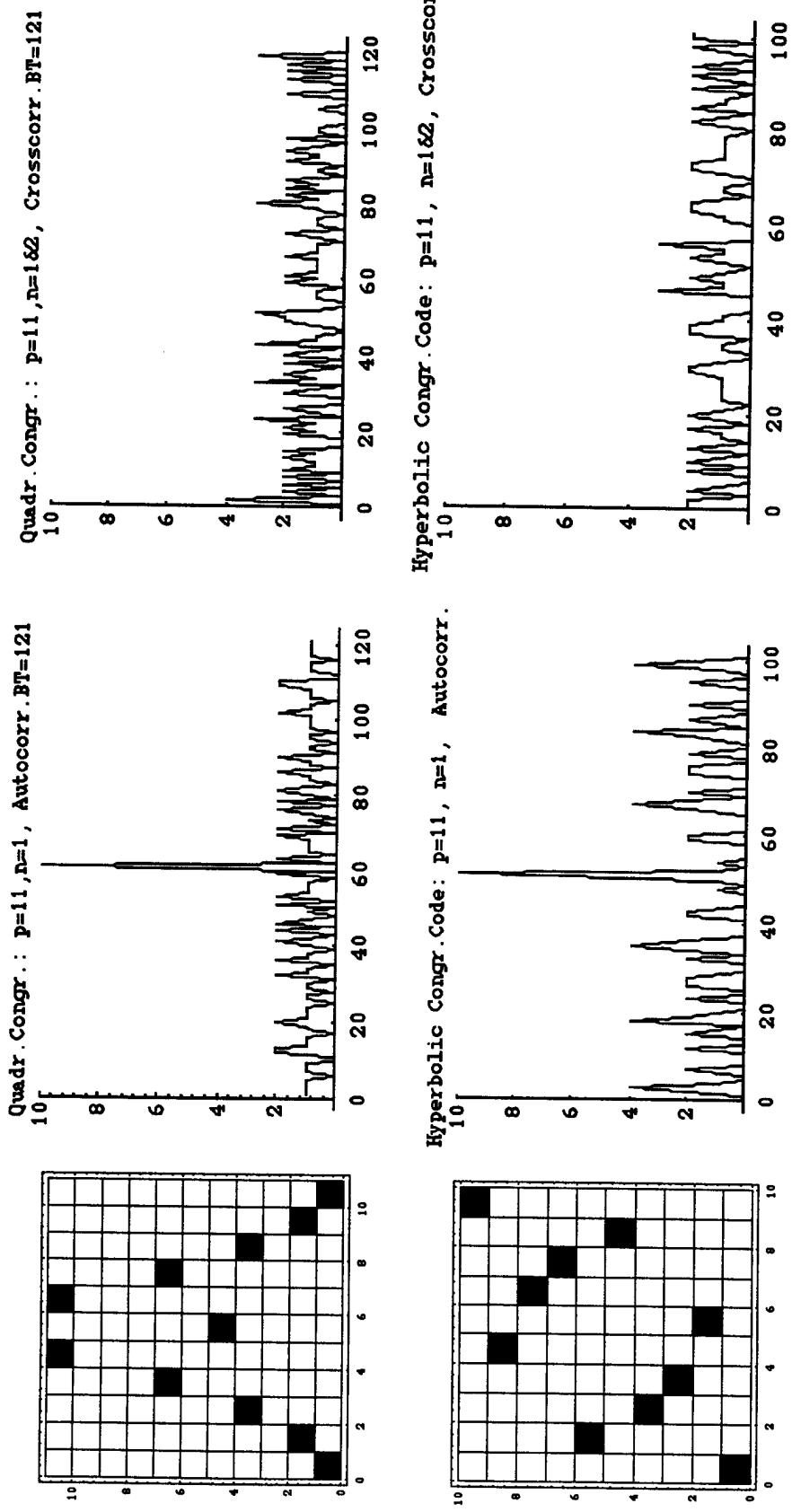


Fig. 9 Auto- and cross-correlations of quadratic congruence<sup>5</sup> and hyperbolic congruence codes.

<sup>5</sup> Maric, S.V., Kostic, Z.I. & Titlebaum, E.R., A new family of optical code sequences for use in spread-spectrum fiber-optic local area networks. *IEEE Trans. Comm.*, 41, 1217-1220, 1993.

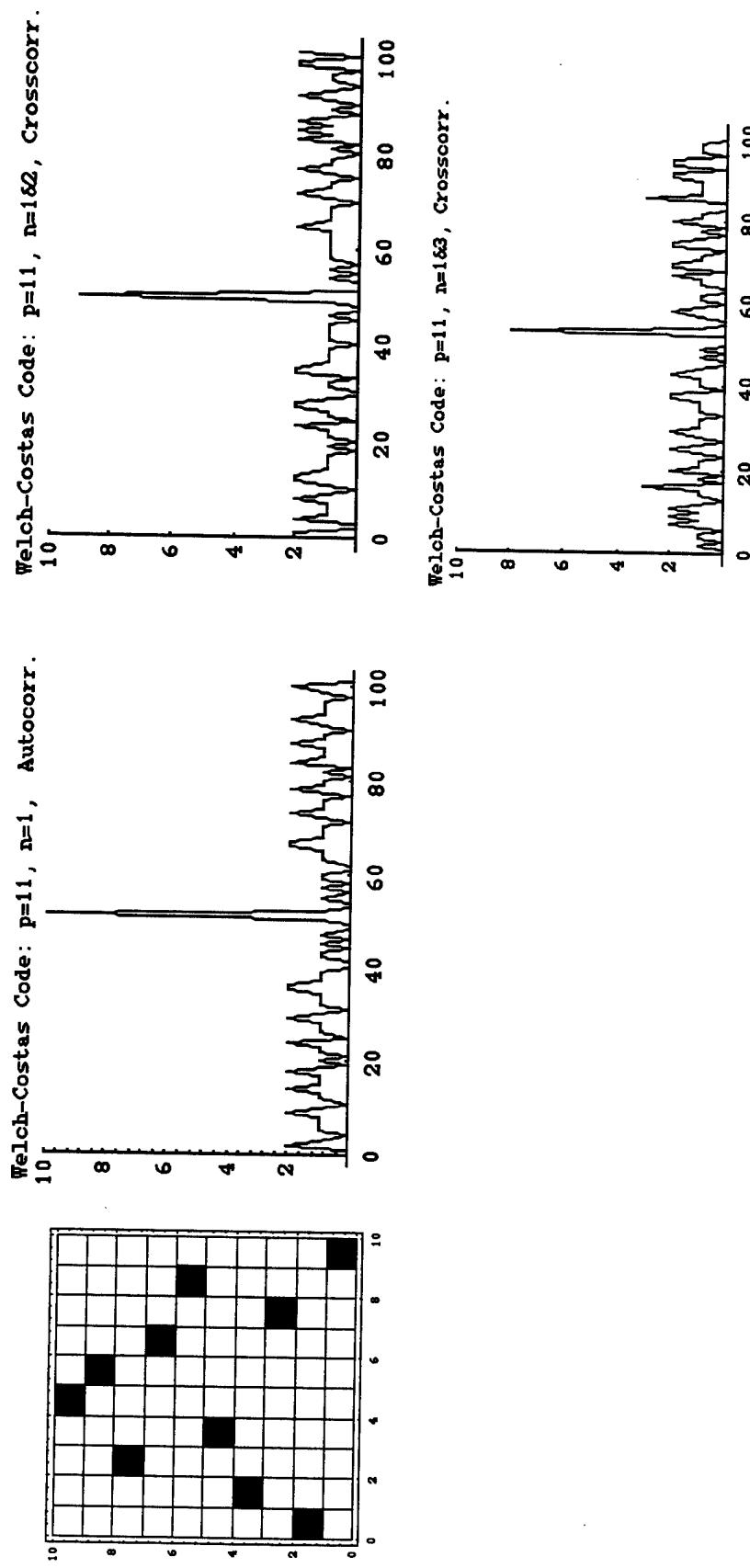


Fig. 10 Auto- and Cross-correlation of Welch-Costas codes<sup>6</sup>. Notice the good auto-correlation but the poor cross-correlation.

<sup>6</sup> Costas, J.P., Time-frequency allocation techniques for active sonar. In *General Electric Technical Information Series*, February, 1970;

---

Costas, J.P., A study of a class of detection waveforms having nearly ideal range-doppler ambiguity properties. *Proc. IEEE*, 72, 996-701, 1984.

Golomb, S.W. & Taylor, H., Construction and properties of Costas arrays. *Proc. IEEE*, 72, 1143-1163, 1984.

Freedman, A. & Levanon, N., Any two  $N \times N$  Costas signals must have at least one common ambiguity sidelobe if  $N > 3$  - a proof. *Proc. IEEE*, 73, 1530-1531, 1985.

Chang, W. & Scarborough, K., Costas arrays with small number of cross-coincidences. *IEEE Trans. Aerospace & Electronic Systems* 25, 109-112, 1989.

Hyperbolic CC/p=11/n=1 Auto-Hit Array   Hyperbolic CC, p=11/n=1&3/Cross-Hit Array

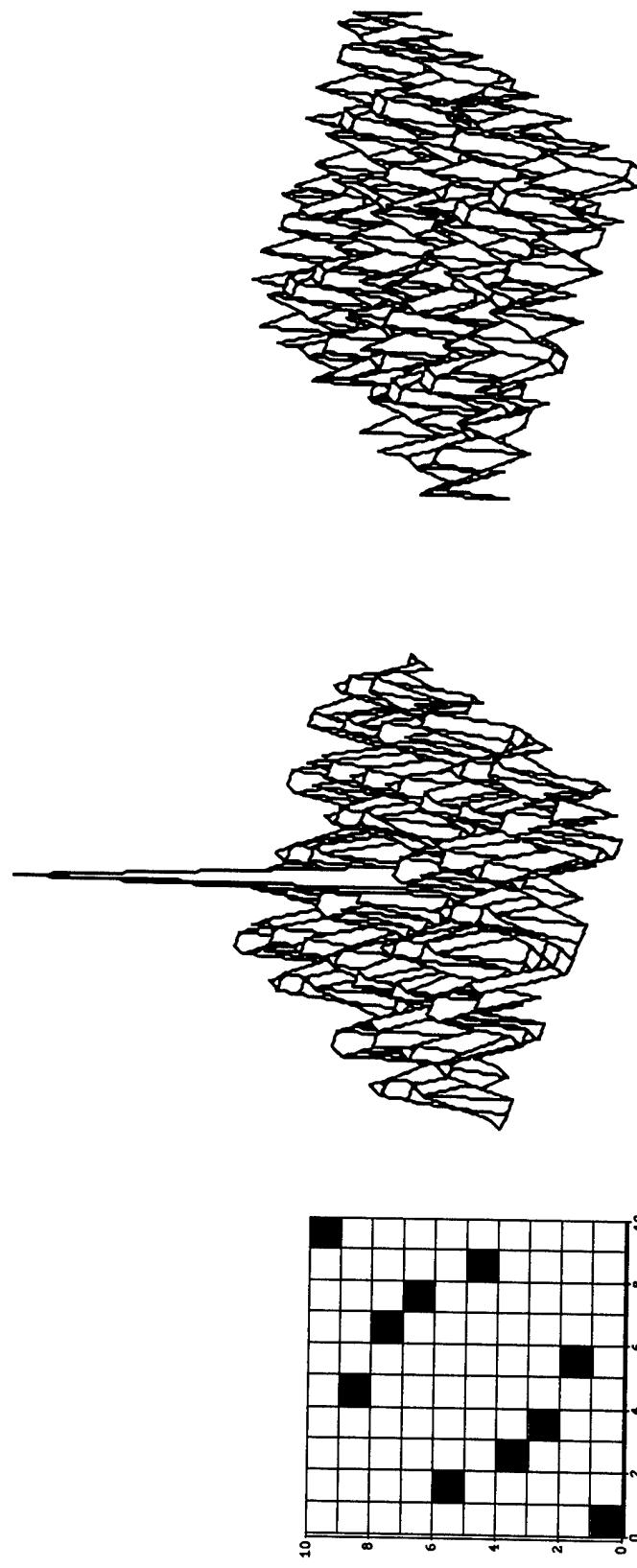


Fig. 11 Auto hit array and Cross hit array for hyperbolic congruence code.

Costas Code  $p=11, n=1$ , Auto-Hit Array

Costas Codes,  $p=11$ , Cross-Hit Array

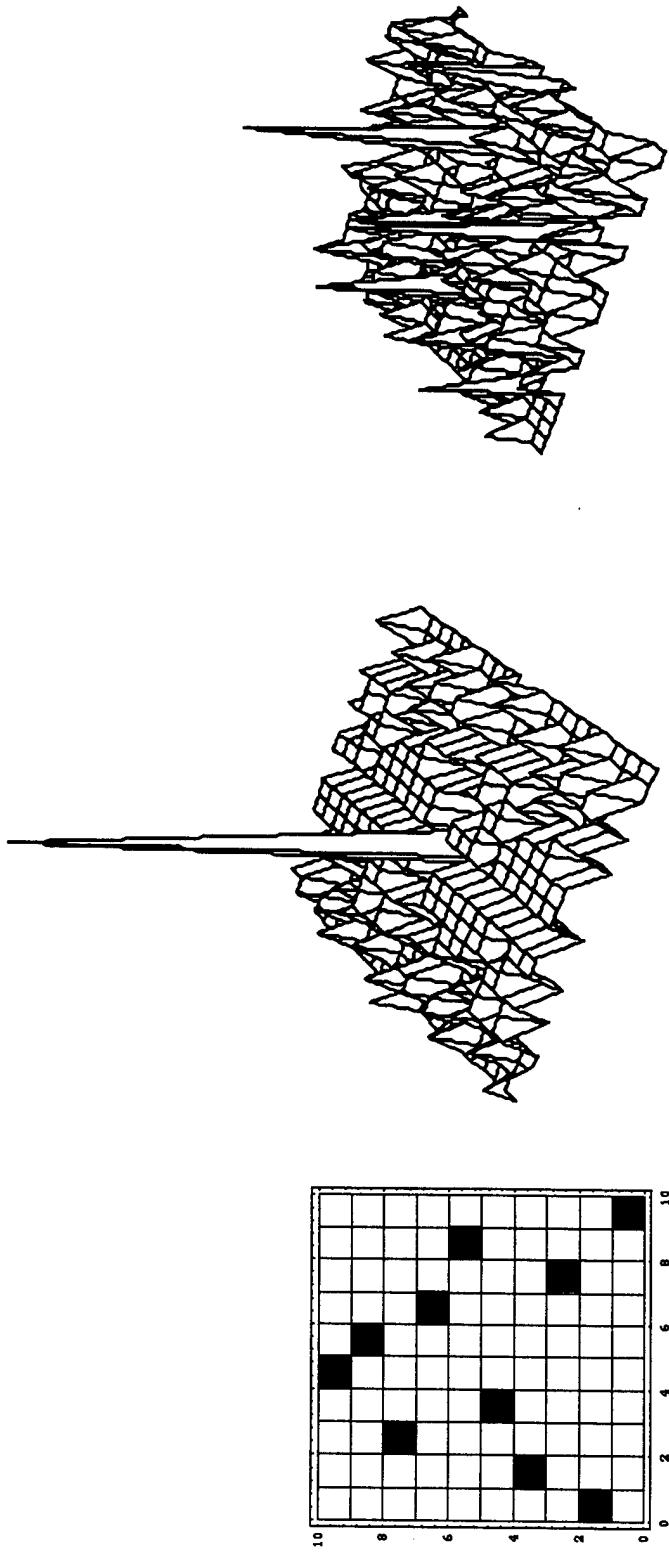
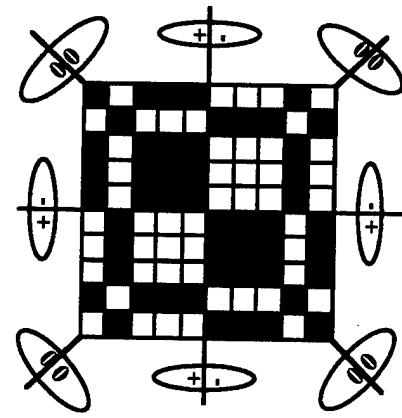


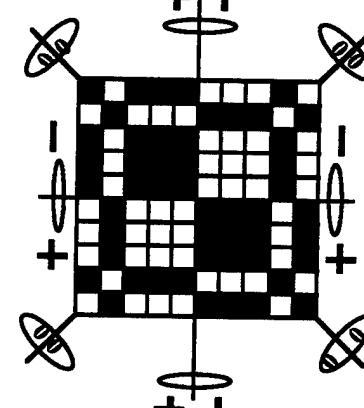
Fig. 12 Auto hit array and Cross hit array for Welch-Costas code. Notice the good Auto hit array, but the poor Cross hit array.

## RESIDUE-NONRESIDUE MATRICES

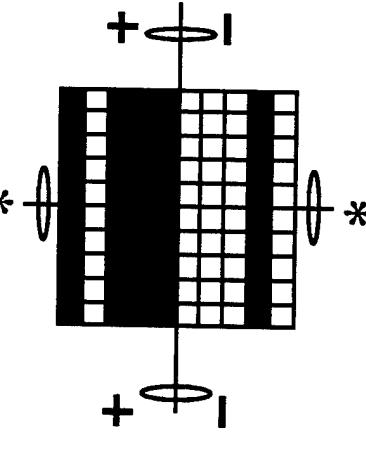
A. Hyperbolic Congruence Codes



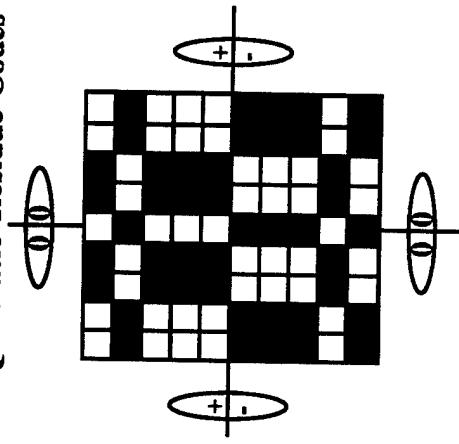
B. Cubic Congruence Codes



C. Quartic Congruence Codes



D. Quadratic Residue Codes



E. Welch-Costas Codes

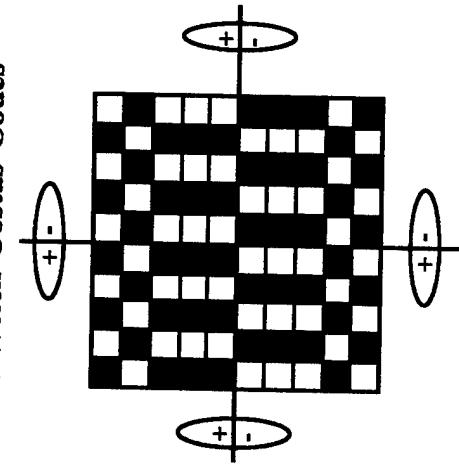


Fig. 13. A: Hyperbolic congruence codes: underlying symmetry is fourfold, and the matrix is equal to its transpose.  
B. Cubic congruence codes: underlying symmetry is fourfold.  
C. Quartic congruence residue codes: twofold symmetry.  
D. Quadratic congruence codes: two-fold symmetry.  
E. Welch-Costas codes: two-fold symmetry.

## INCREASING/DECREASING POLYNOMIAL INDICES

## Cubic Congruence Codes

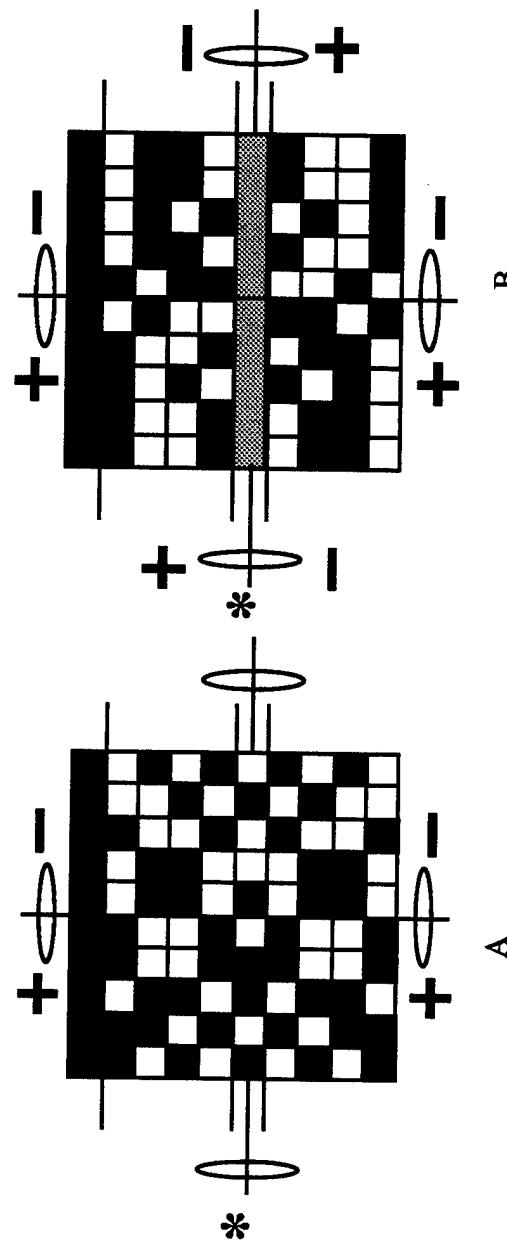


Fig. 14 Matrix representation as increasing (black) or decreasing (white) generator polynomial index (a).  
 A: Cubic congruence codes; reverse symmetry (left to right) and an exact symmetry.  
 B: Quartic congruence codes; reverse symmetry (left to right) and reverse symmetry ('top to bottom')

## INCREASING/DECREASING POLYNOMIAL INDICES

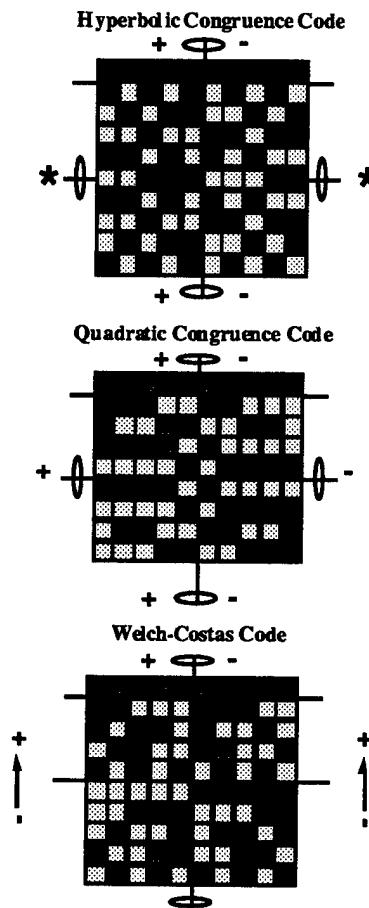


Fig. 15 The full 10 hyperbolic congruence, quadratic congruence and Welch-Costas codes are represented as an increasing (dark shading) or decreasing (light shading) generator polynomial index ( $\alpha$ ). The patterns indicate the *sequential* increase or decrease of the  $\alpha$  input to a shift register. The significant discovery is that there are symmetrical patterns corresponding to each orthogonal code, e.g.:

- (1) In the case of the hyperbolic congruence codes and neglecting the top row: (a) there is a *reverse* symmetry about the y-axis; and (b) there is an *exact* symmetry about the x-axis.
- (2) In the case of the quadratic congruence codes and neglecting the top row: (a) there is a *reverse* symmetry about the y-axis; and (b) there is a *reverse* symmetry about the x-axis.
- (3) In the case of the Welch-Costas codes: (a) neglecting the top (10th) row, there is a *reverse* symmetry about the y-axis; and (b) there is a *reversal and a displacement* over the x-axis.

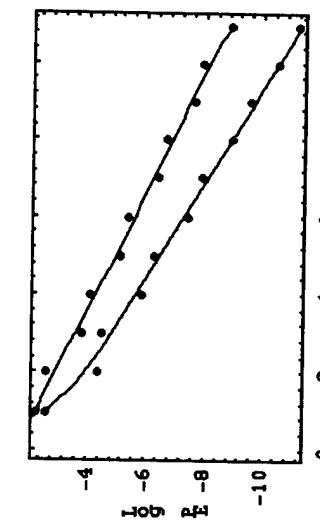


Fig. 16

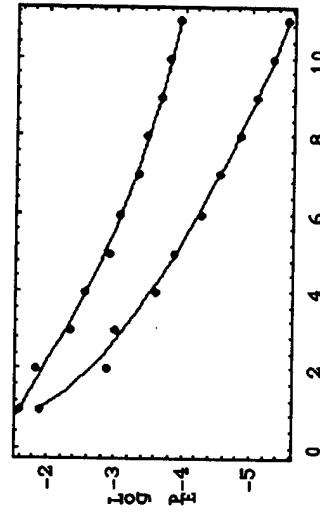


Fig. 17

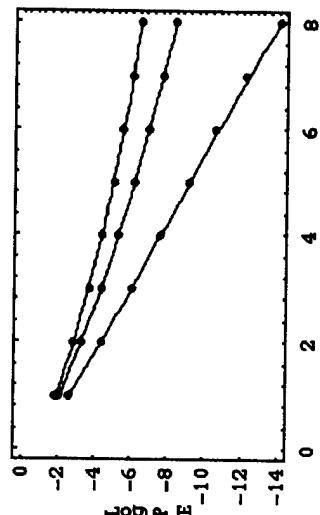


Fig. 18

Fig. 16. Bit error probability for an OOC with  $F = 1000$ ,  $M = 10,30,50$  (lower to upper) and  $K = Th$ . After Azizoglu et al<sup>7</sup>.

Fig. 17. Upper and lower bounds to the worst case error probability for the  $\lambda = 2$  code,  $M = 50$ . After Ref<sup>7</sup>.

Fig. 18. Upper and lower bounds to the worst case error probability for the  $\lambda = 2$  code,  $M = 10$ . After Ref<sup>7</sup>.

<sup>7</sup> Azizoglu, M.Y., Salehi, J.A. & Li, Y., On the performance of fiber-optic CDMA systems. *Proc. IEEE GLOBECOM 1990*, San Diego, CA, Dec 1990, pp. 1861-1865;  
Azizoglu, M.Y., Salehi, J.A. & Li, Y., Optical CDMA via temporal codes. *IEEE Trans. Comm.*, 40, 1162-1170, 1992.

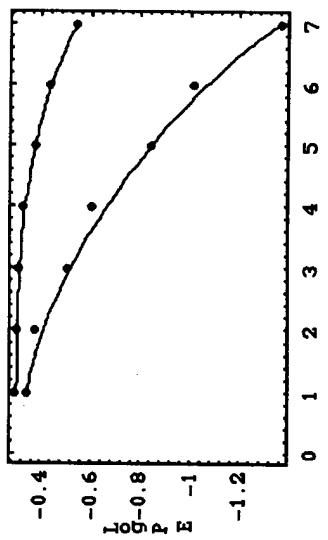


Fig. 19

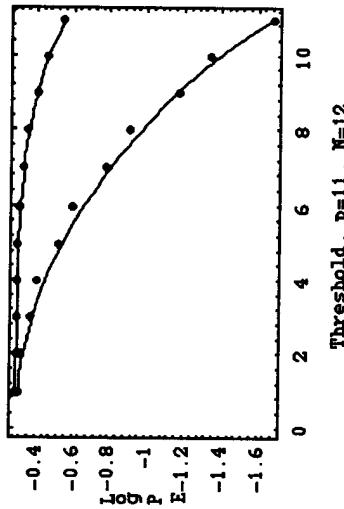


Fig. 20

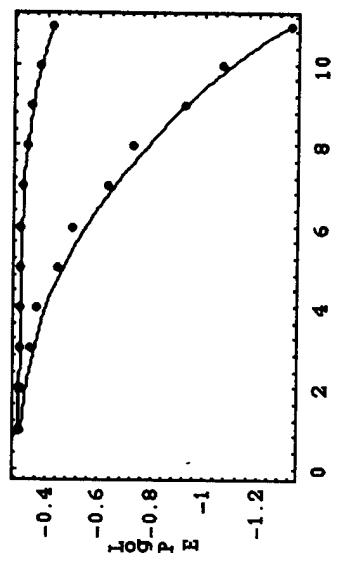


Fig. 21

Fig. 19. Error probability versus matched filter threshold for  $p = 7, N = 8$ .  $C_{gc}$  (lower),  $C'_{gc}$  (upper). After Kostic & Titlebaum<sup>8</sup>.

Fig. 20. Error probability versus matched filter threshold for  $p = 11, N = 12$ .  $C_{gc}$  (lower),  $C'_{gc}$  (upper). After Ref<sup>8</sup>.

Fig. 21. Error probability versus matched filter threshold for  $p = 11, N = 14$ .  $C_{gc}$  (lower),  $C'_{gc}$  (upper). After Ref<sup>8</sup>.

<sup>8</sup> Kostic, Z. & Titlebaum, E.I., The design and performance analysis for several new classes of codes for optical synchronous CDMA and for arbitrary-medium time-hopping synchronous CDMA communication systems. *IEEE Trans. Comm.*, 42, 2608-2617, 1994.

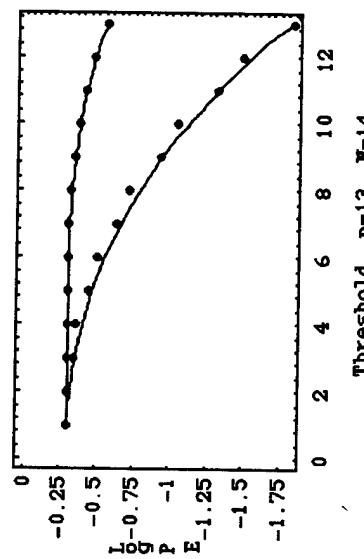


Fig. 22

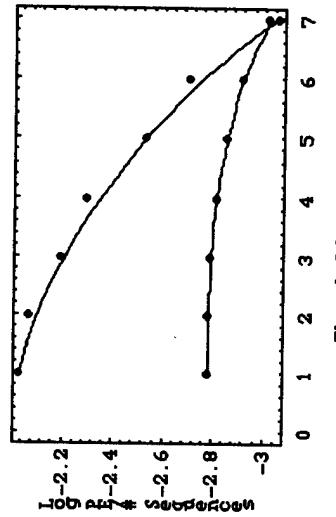


Fig. 23

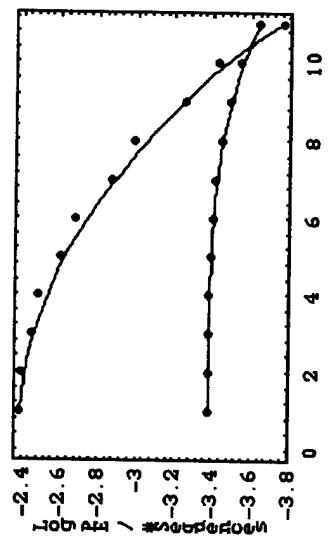


Fig. 24

Fig. 22. Error probability versus matched filter threshold for  $p = 13, N = 14$ .  $C_{gc}$  (lower),  $C'_{gc}$  (upper). After Ref<sup>8</sup>.

Fig. 23. Log PE/number of sequences in set as a function of threshold.  $p = 7, N = 8$ .  $C_{gc}$  (upper);  $C'_{gc}$  (lower).

Fig. 24. Log PE/number of sequences in set as a function of threshold.  $p = 11, N = 12$ .  $C_{gc}$  (upper);  $C'_{gc}$  (lower).

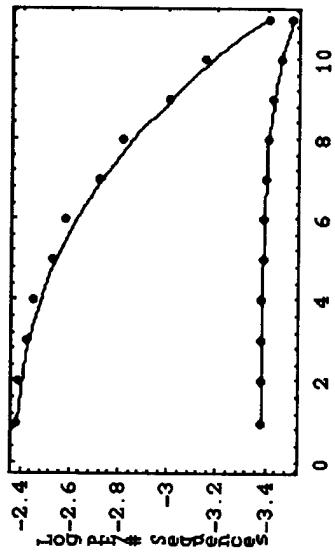


Fig. 25

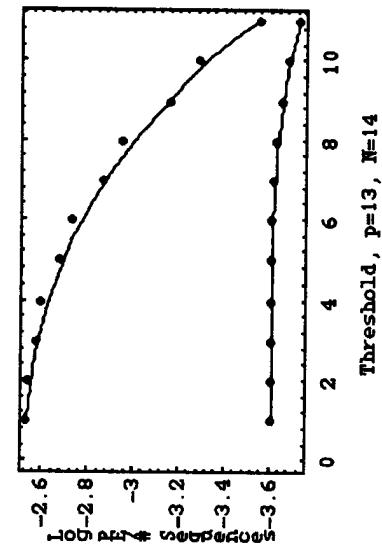


Fig. 26

Fig. 25. Log PE/number of sequences in set as a function of threshold.  $p = 11, N = 14$ .  $C_{gc}$  (upper);  $C'_{gc}$  (lower). After Ref<sup>8</sup>.

Fig. 26. Log PE/number of sequences in set as a function of threshold.  $p = 13, N = 14$ .  $C_{gc}$  (upper);  $C'_{gc}$  (lower).

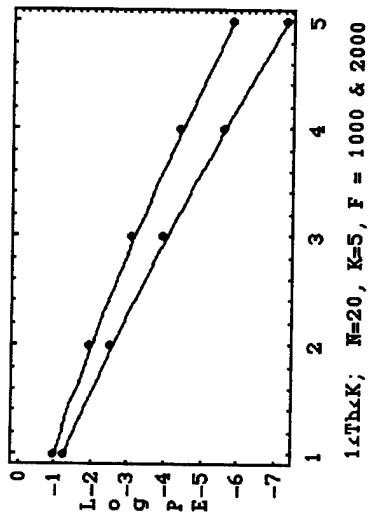


Fig. 27

Fig. 27. Upper bound on bit error rate (chip synchronous):  $N = 20, K = 5$ . Upper,  $F = 1000$ ; Lower,  $F = 2000$ .

Fig. 28. Upper bound on bit error rate (chip synchronous):  $N = 10, F = 1000, K = 1, 3, 5, 7 \& 9$ .

Fig. 29. Upper bound on bit error rate (chip synchronous):  $K = 5, F = 1000, N = 10, 30 \text{ and } 50$ , left-to-right. This is the case of the number of slots in a superframe,  $F$ , being much larger than the number of users,  $N$ . In the case, increasing the number of users results in a major reduction in performance. After Ref<sup>9</sup>, Fig. 3(c).

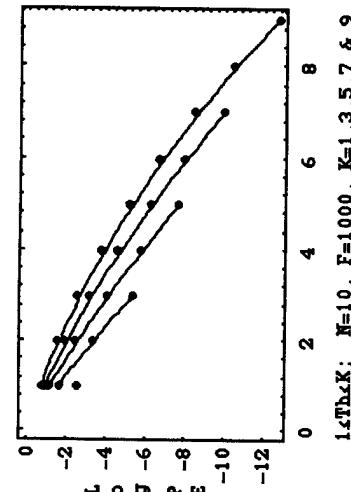


Fig. 28

Fig. 27. Upper bound on bit error rate (chip synchronous):  $N = 20, K = 5$ . Upper,  $F = 1000$ ; Lower,  $F = 2000$ . A reduction in the total number of chips per frame,  $F$ , i.e., slots available, results in a reduction in performance. After Salehi & Brackett<sup>9</sup>, Fig. 3(a).

Fig. 28. Upper bound on bit error rate (chip synchronous):  $N = 10, F = 1000, K = 1, 3, 5, 7, 9$ , left-to-right. Although increasing the number of pulses in the code,  $K$ , ostensibly results in a reduction in performance, the possibility of raising the receiver threshold actually results in a considerable increase in performance. After Ref<sup>9</sup>, Fig. 3(b).

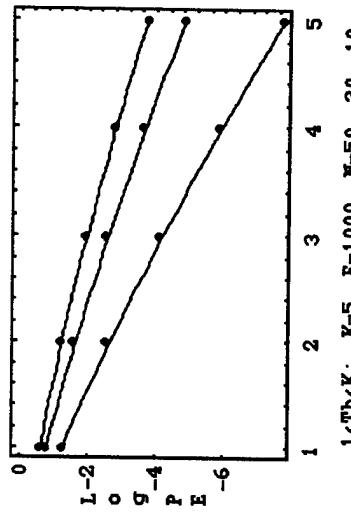


Fig. 29

Fig. 27. Upper bound on bit error rate (chip synchronous):  $N = 20, K = 5$ . Upper,  $F = 1000$ ; Lower,  $F = 2000$ . A reduction in the total number of chips per frame,  $F$ , i.e., slots available, results in a reduction in performance. After Salehi & Brackett<sup>9</sup>, Fig. 3(a).

Fig. 28. Upper bound on bit error rate (chip synchronous):  $N = 10, F = 1000, K = 1, 3, 5, 7, 9$ , left-to-right. Although increasing the number of pulses in the code,  $K$ , ostensibly results in a reduction in performance, the possibility of raising the receiver threshold actually results in a considerable increase in performance. After Ref<sup>9</sup>, Fig. 3(b).

Fig. 29. Upper bound on bit error rate (chip synchronous):  $K = 5, F = 1000, N = 10, 30 \text{ and } 50$ , left-to-right. This is the case of the number of slots in a superframe,  $F$ , being much larger than the number of users,  $N$ . In the case, increasing the number of users results in a major reduction in performance. After Ref<sup>9</sup>, Fig. 3(c).

<sup>9</sup> Salehi, J.A. & Brackett, C.A., Code division multiple-access techniques in optical fiber networks - Part II: Systems performance analysis. *IEEE Trans. Comm.*, 37, 834-842, 1989.

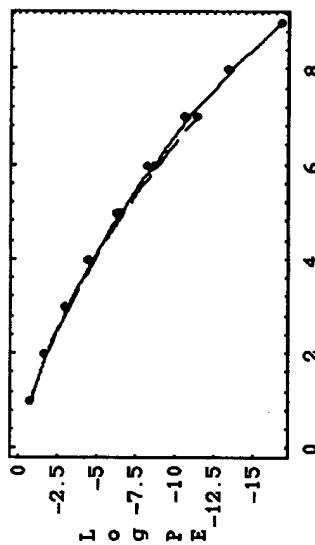


Fig. 30

Fig. 30. Upper bounds on bit error rate (chip synchronous) with optical hard limiter:  $F = 1000, N = 10$  and  $K = 10$  and  $K = 9$  (upper, line),  $K = 7$  (lower, hatched),  $K = 6$  (upper, line). Compare with Fig. 28 showing that hard-limiting increases performance by approx. 1.5 orders of magnitude. After<sup>9</sup>, Fig. 6(b).

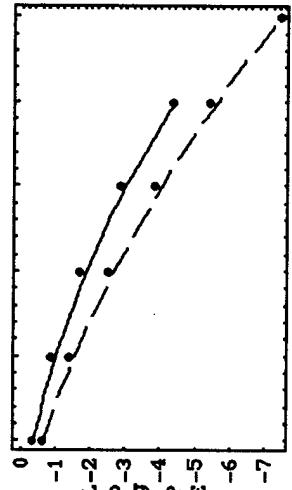


Fig. 31

Fig. 31. Upper bounds on bit error rate (chip synchronous) with optical hard-limiter:  $F = 1000, N = 50$  and  $K = 5$  (upper, line) and  $N = 30$  and  $K = 6$  (lower, hatched). Compare with Fig. 27. After Ref<sup>9</sup>, Fig. 6(a).

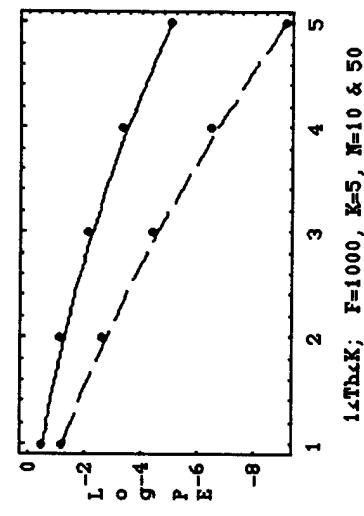


Fig. 32

Fig. 32. Upper bounds on bit error rate (chip synchronous) with optical hard limiter:  $F = 1000, K = 5$  and  $N = 10$  (lower, hatched),  $N = 50$  (upper, line). Compare with Fig. 29. After Ref<sup>9</sup>, Fig. 6(c).

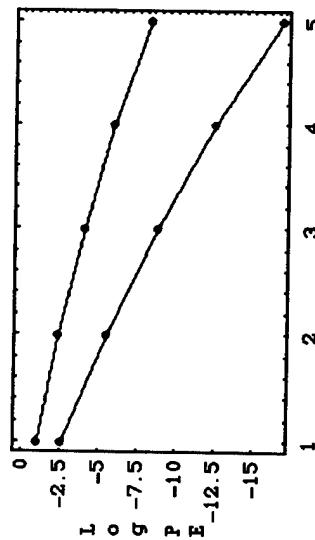


Fig. 33

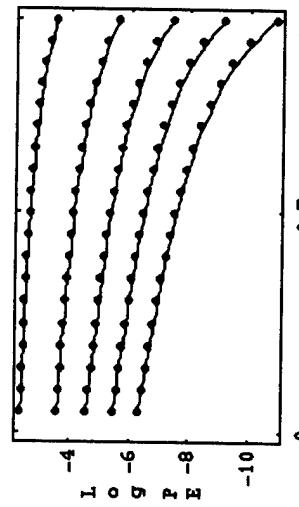


Fig. 34

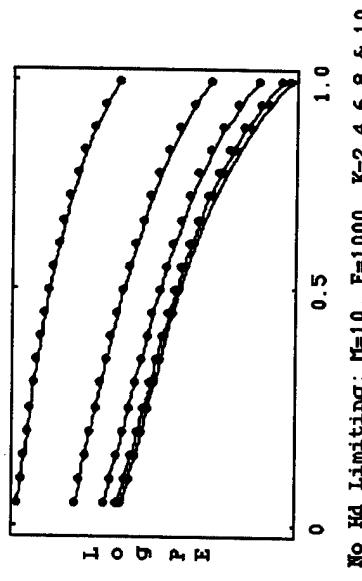


Fig. 35

Fig. 33. Upper bounds on bit error rate (chip asynchronous) with optical hard limiter:  $F = 1000, K = 5$  and  $N = 10$  (lower line) and  $N = 50$  (upper line). With hard-limiting the system performance increases by 4-5 orders of magnitude. After Ref<sup>9</sup>, Fig. 7.

Fig. 34.  $\lambda=2$ , Optical Orthogonal Code, without hard limiting. Dependence of the error probability on the parameter  $c = q_i/(K^2/2F)$  and  $M \leq l(F-1)(F-2)/l(K(K-1)(K-2))$ .  $M=10, F=1000, K=2, 4, 6, 8 \& 10$ , top to bottom. After Ref<sup>10</sup>, Fig. 4(a).

Fig. 35.  $\lambda=2$ , Optical Orthogonal Code, without hard limiting. Dependence of the error probability on the parameter  $c = q_i/(K^2/2F)$  and  $M \leq l(F-1)(F-2)/l(K(K-1)(K-2))$ .  $M=50, F=1000, K=2, 4, 6, 8 \& 10$ , top to bottom. After Ref<sup>10</sup>, Fig. 4(b).

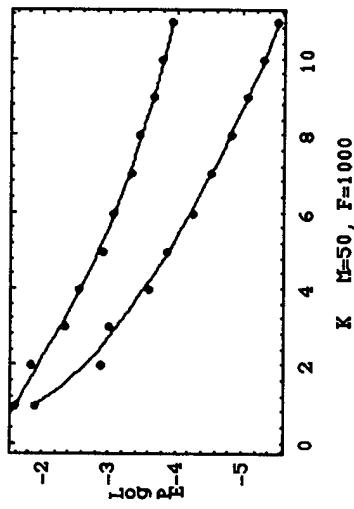


Fig. 36

Fig. 36. Upper and lower bounds to the worst case error probability for the  $\lambda = 2$  code,  $M = 50$ , with hard limiting. After Ref<sup>10</sup>, Fig. 5 upper.

Fig. 37. Upper and lower bounds to the worst case error probability for the  $\lambda = 2$  code,  $M = 10$ , with hard limiting. After Ref<sup>10</sup>, Fig. 5 lower.

Fig. 38. Bit error probability for an OOC ( $\lambda=1$ ) with  $F = 1000$ ,  $M = 10,30,50$  (lower to upper) and  $K = Th$ . After Ref<sup>10</sup>, Fig. 2.

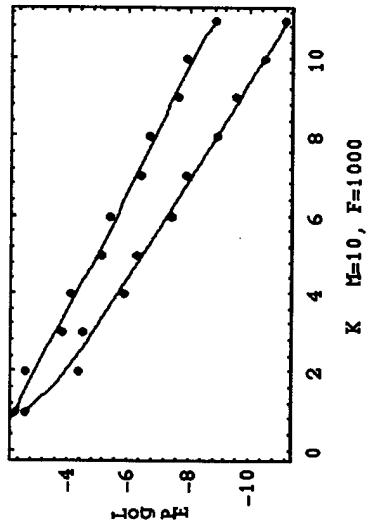


Fig. 37

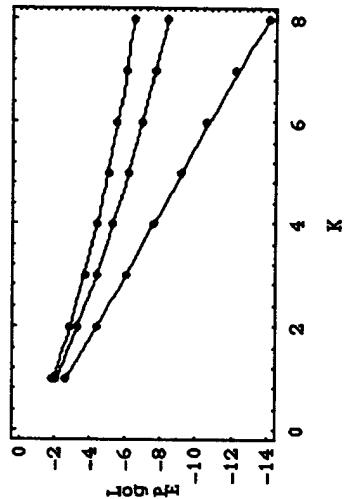


Fig. 38

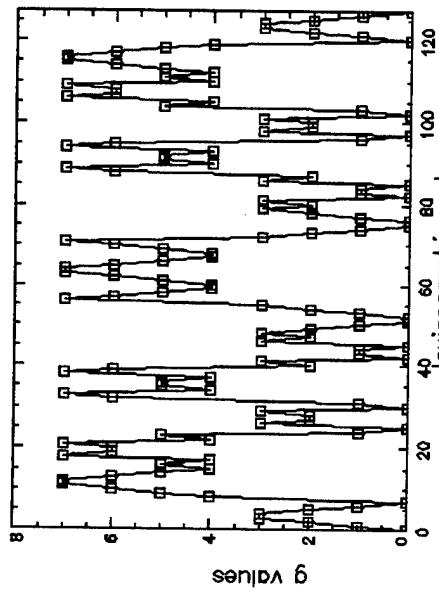


Fig. 39. The  $g$ -values of the Lexicographic-Greedy Code (7,4,3) of Appendix Table 7.

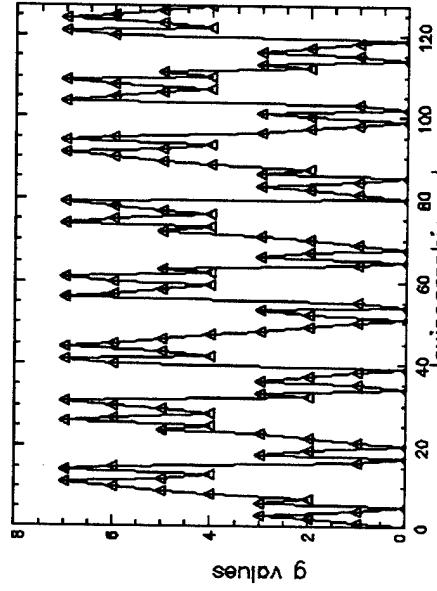


Fig. 40. The  $g$ -values of the Lexicographic-Greedy Code (7,4,3) of Appendix Table 8.

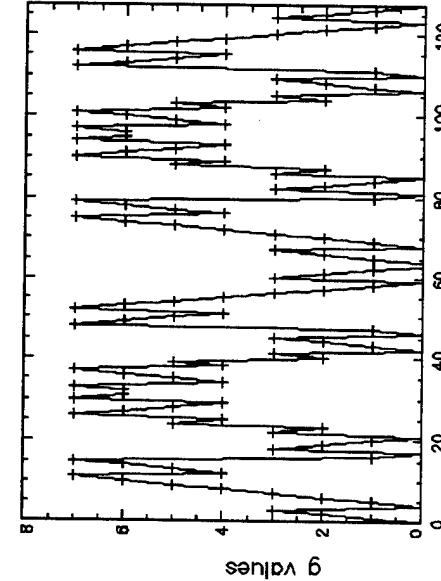


Fig. 41. The  $g$ -values of the Lexicographic-Greedy Code (7,4,3) of Appendix Table 9.

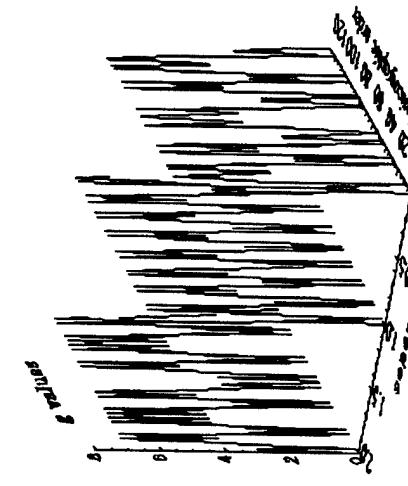


Fig. 42. The  $g$ -values of the Lexicographic-Greedy Codes (7,4,3) of Appendix Tables 7-9 and shown separately in Figs 39-41, above.

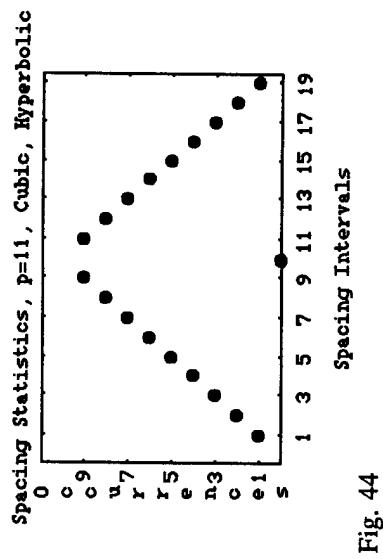


Fig. 44

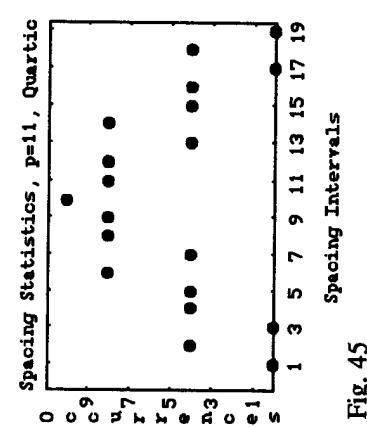


Fig. 45

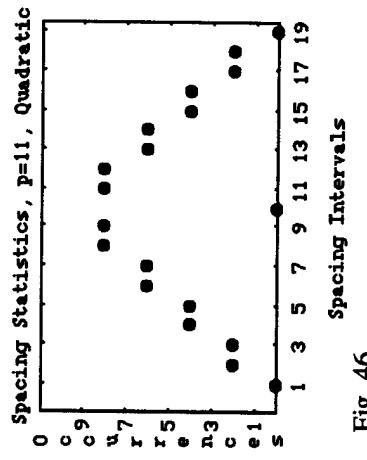


Fig. 46

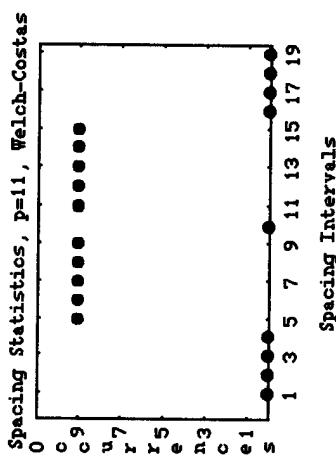


Fig. 47

Fig. 44. The number of occurrences of specific spacing intervals for the cubic and hyperbolic codes,  $p = 11$ .

Fig. 45. The number of occurrences of specific spacing intervals for the quartic codes,  $p = 11$ .

Fig. 46. The number of occurrences of specific spacing intervals for the quadratic congruence codes,  $p = 11$ .

Fig. 47. The number of occurrences of specific spacing intervals for the Welch-Costas codes,  $p = 11$ .

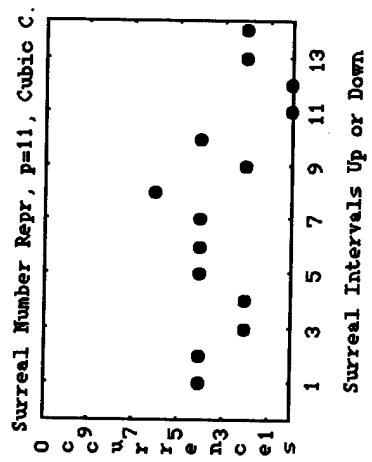


Fig. 48

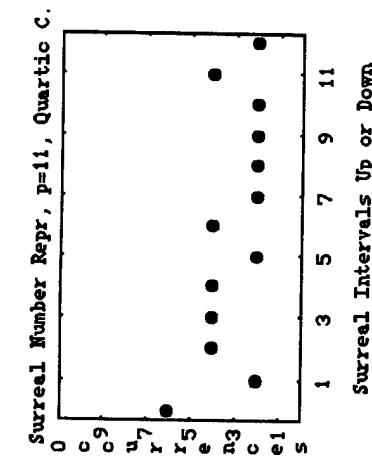


Fig. 49

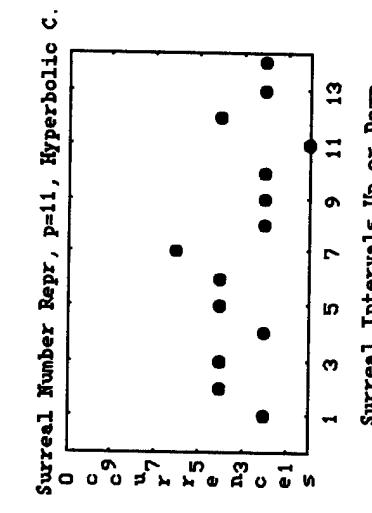


Fig. 50

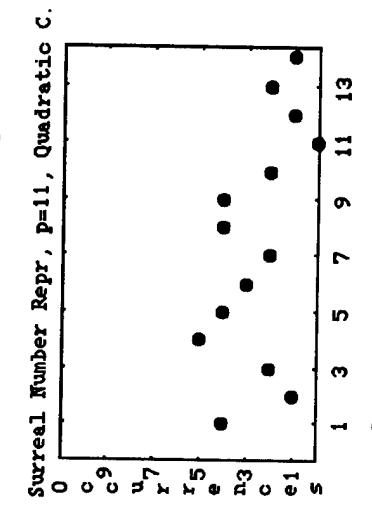


Fig. 51

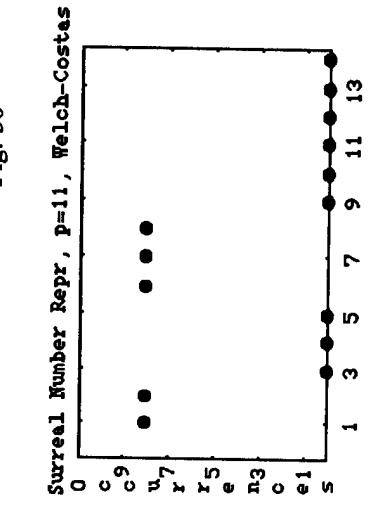


Fig. 52

Fig. 48. Cubic Congruence Codes, surreal number representation.

Fig. 49. Quartic Codes, surreal number representation.

Fig. 50. Hyperbolic congruence codes, surreal number representation.

Fig. 51. Quadratic congruence codes, surreal number representation.

Fig. 52. Welch-Costas codes, surreal number representation.

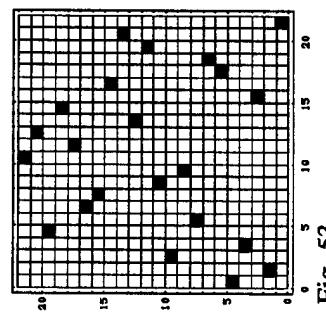


Fig. 53



Fig. 54



Fig. 55

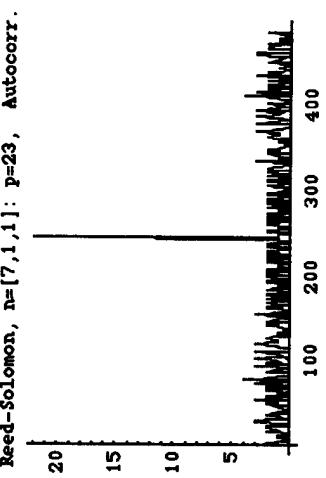


Fig. 56

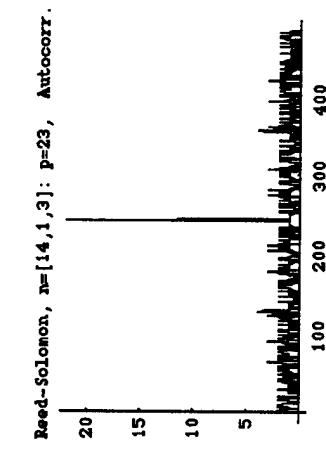


Fig. 57

Fig. 53. Reed-Solomon code:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 5 & 5^2 & 5^3 & 5^4 & \cdots & 5^{22} = 1 \\ 5^2 & 5^4 & 5^6 & 5^8 & \cdots & 1 \\ 5^3 & 5^6 & 5^9 & 5^{12} & \cdots & 1 \\ 5^4 & 5^8 & 5^{12} & 5^{16} & \cdots & 1 \\ 5^5 & 5^{10} & 5^{15} & 5^{20} & \cdots & 1 \end{bmatrix}$ ,  $n = [0, 1, 0, 0]$ ,  $p = 23$ .

Fig. 54. Reed-Solomon code:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 5 & 5^2 & 5^3 & 5^4 & \cdots & 5^{22} = 1 \\ 5^2 & 5^4 & 5^6 & 5^8 & \cdots & 1 \\ 5^3 & 5^6 & 5^9 & 5^{12} & \cdots & 1 \\ 5^4 & 5^8 & 5^{12} & 5^{16} & \cdots & 1 \\ 5^5 & 5^{10} & 5^{15} & 5^{20} & \cdots & 1 \end{bmatrix}$ ,  $n = [0, 1, 1, 0]$ ,  $p = 23$ .

Fig. 55. Reed-Solomon code, Example 1:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 5 & 5^2 & 5^3 & 5^4 & \cdots & 1 \\ 5^2 & 5^4 & 5^6 & 5^8 & \cdots & 1 \\ 5^3 & 5^6 & 5^{12} & 5^{18} & \cdots & 1 \end{bmatrix}$ ,  $n = [7, 1, 1]$ ,  $p = 2$ .

Fig. 53. Reed-Solomon code:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$ ,  $n = [0, 1, 0, 0]$ ,  $p = 23$ .

Fig. 54. Reed-Solomon code:  $c = nG$ ,  $G = \begin{bmatrix} 5 & 5^2 & 5^3 & \dots & 5^{22} = 1 \\ 5^2 & 5^4 & 5^6 & \dots & 1 \\ 5^3 & 5^6 & 5^9 & \dots & 1 \end{bmatrix}$ ,  $n = [0, 1, 1, 0]$ ,  $p = 23$ .

Fig. 54. Reed-Solomon code:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$ ,  $n = [0, 1, 1, 0]$ ,  $p = 23$ .

Fig. 55. Reed-Solomon code, Example 1:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 5 & 5^2 & 5^3 & 5^4 & \dots & 1 \\ 5^2 & 5^4 & 5^6 & 5^8 & \dots & 1 \end{bmatrix}$ ,  $n = 7, l, l$ ,  $p = 23$ .

Fig. 55. Reed-Solomon code, Example 1:  $C = nG$ ,  $G \equiv$

Fig. 56. Reed-Solomon code, Example 2:  $c = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 5 & 5^2 & 5^3 & \cdots & 1 \\ 5^2 & 5^4 & 5^6 & \cdots & 1 \end{bmatrix}$ ,  $n = [14, 1, 3]$ ,  $p = 23$ .

Fig. 5.6 Reed-Solomon code. Example 2:  $c \equiv mg$   $G = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$

## 1.2 Code Bounds

Lexicographic-Greedy error-correcting codes are defined as follows. Every linear code  $\mathbf{C}$  with minimum distance at least  $d$  and covering radius at most  $d-1$  is a  $\mathbf{B}$ -greedy code of designed distance  $d$  for some ordered basis  $\mathbf{B}$ . Any ordered basis for  $\mathbf{B}$  may be chosen whose first  $k$  vectors are a basis of  $\mathbf{C}$  where  $k$  is the dimension of  $\mathbf{C}$ . The fact that a  $\mathbf{B}$ -greedy code of designed distance  $d$  has covering radius at most  $d-1$  implies that  $\mathbf{B}$ -greedy codes attain the *Varshamov-Gilbert*<sup>11</sup> bound for binary linear codes. This month we demonstrate that the standard arrays of Lexicographic-Greedy codes are ordered in cosets defined by the coset  $g$ -values. The coset leaders may, or may not, be the basis vectors defining the code. We have also discovered symmetries underlying the lexicographic ordering of the  $g$ -values of these codes.

If an  $(n,k)$  linear<sup>12</sup> code is considered with generator matrix  $\mathbf{G}$  and parity check matrix  $\mathbf{H}$ , then a transmitted code word,  $\mathbf{v}$ , is defined as:

$$\mathbf{v} = (v_0, v_1, \dots, v_{n-1}),$$

and a received vector as:

$$\mathbf{r} = (r_0, r_1, \dots, r_{n-1}).$$

The error vector,  $\mathbf{e}$ , is then defined as:

$$\begin{aligned} \mathbf{e} &= \mathbf{r} + \mathbf{v} \\ &= (e_0, e_1, \dots, e_{n-1}). \end{aligned}$$

A syndrome,  $\mathbf{s}$ , is defined as:

$$\begin{aligned} \mathbf{s} &= \mathbf{r} \bullet \mathbf{H}^T \\ &= (s_0, s_1, \dots, s_{n-k-1}) \end{aligned}$$

where  $\mathbf{H}$  is a parity check matrix:

---

<sup>11</sup> The *Varshamov-Gilbert bound* states that for any minimum weight  $d$  and redundancy  $m$  there is a code of length  $n$  that can correct  $[(d-1)/2]$  errors and has dimension greater than or equal to  $n - m$ .

<sup>12</sup> Linear block codes are defined in terms of generator and parity-check matrices. Specifically, a linear block code of length  $n$  and  $2^k$  code words is called a linear  $(n,k)$  code iff its  $2^k$  code words form a  $k$ -dimensional subspace of the vector space of all the  $n$ -tuples over the field GF(2). Stated differently, a binary block code is linear iff the modulo-2 sum of two code words is also a code word.

$$\begin{aligned}
 H &= [I_{n-k} \ P^T] \\
 &= \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & p_{00} & p_{10} & \dots & p_{k-1,0} \\ 0 & 1 & 0 & \dots & 0 & p_{01} & p_{11} & \dots & p_{k-1,1} \\ 0 & 0 & 1 & \dots & 0 & p_{02} & p_{12} & \dots & p_{k-1,2} \\ \vdots & & \vdots & & \vdots & & & & \\ 0 & 0 & 0 & \dots & 0 & p_{0,n-k-1} & p_{1,n-k-1} & \dots & p_{k-1,n-k-1} \end{bmatrix}
 \end{aligned}$$

which provides the elements for the matrix  $P$  as in:

$$H = [I_{n-k} \ P^T],$$

where  $I_{n-k}$  is the identity matrix.

We shall refer to this syndrome and parity check matrix as the systematic-syndrome and the systematic parity check matrix, or the S-syndrome and the S-parity check matrix. These are the conventional syndromes and parity-check matrices.

$s = 0$  if and only if  $r$  is a code word, and  $s \neq 0$  if and only if  $r$  is not a code word. A further eventuality is that  $s = r \cdot H^T = 0$ , but  $r$  is identical to a nonzero code word. This is the case of an undetectable error pattern.

According to these definitions, the S-syndrome is:

$$\begin{aligned}
 s_0 &= r_0 + r_{n-k}p_{00} + r_{n-k+1}p_{10} + \dots + r_{n-1}p_{k-1,0} \\
 s_1 &= r_1 + r_{n-k}p_{01} + r_{n-k+1}p_{11} + \dots + r_{n-1}p_{k-1,1} \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 s_{n-k-1} &= r_{n-k-1} + r_{n-k}p_{0,n-k-1} + r_{n-k+1}p_{1,n-k-1} + \dots + r_{n-1}p_{k-1,n-k-1}
 \end{aligned}$$

Returning now to the Greedy codes, an index  $m$  is defined as the smallest integer such that

$$g(z) \leq 2^m - 1 \quad \text{for all } z \text{ in } F_2^m.$$

In the case of the (5,3) lexicographic code, the highest  $g$  value is 7. Therefore, the index  $m$  is 3. The  $g$ -value for every  $z$  vector in  $F_2^m$  can be made a vector by taking its base 2 description. Therefore, the  $g$  values may be regarded as maps<sup>13</sup>:

$$g : F_2^m \rightarrow F_2^m.$$

<sup>13</sup> Brualdi, R.A. & Pless, V.S., Greedy codes, *J. Combinatorial Theory, Series A* 64, 1993, 1-30.

In the case of the lexicographic code considered, this mapping is:

$$g : F_2^5 \rightarrow F_2^3$$

In the case of the  $n = 5, k = 2, d = 3$  Lexicographic code given in Table 1 of the March Report, a parity check matrix for the  $g$ -values of vectors in  $F_2$  is:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix},$$

where we have used the observation<sup>13</sup> that

$$\mathbf{H} = [g(e_5)g(e_4)g(e_3)g(e_2)g(e_1)],$$

reflecting:

$e_i$	g values	g values as vectors or $g(e_i)$
y1= { 0, 0, 0, 0, 1}	1	( 0 0 1 )
y2= { 0, 0, 0, 1, 0}	2	( 0 1 0 )
y3= { 0, 0, 1, 0, 0}	3	( 0 1 1 )
y4= { 0, 1, 0, 0, 0}	4	( 1 0 0 )
y5= { 1, 0, 0, 0, 0}	5	( 1 0 1 )

We shall refer to this syndrome and parity check matrix as the Lexicographic or L-syndrome and the Lexicographic or L-parity check matrix, in comparison with the previously defined conventional S-syndrome and S-parity check matrix.

As an example, if the  $z$  vector {0,0,0,1,1} from that Table (i.e., a  $z$  vector which is *not* a code word) is considered, then its  $g$  value, 5, is its L-syndrome relative to the parity check matrix. This can be seen as follows. The L-syndrome for that Lexicographic code is:

$$s = (s_0, s_1, s_2) = (r_0, r_1, r_2, r_3, r_4) \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = r \bullet H^T \text{ or}$$

$$s = (s_0, s_1, s_2) = (r_0, r_1, r_2, r_3, r_4) \begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \\ p_{30} & p_{31} & p_{32} \\ p_{40} & p_{41} & p_{42} \end{bmatrix}$$

Therefore, as:

$$s_0 = r_0 \times p_{00} + r_1 \times p_{10} + r_2 \times p_{20} + r_3 \times p_{30} + r_4 \times p_{40}$$

$$s_1 = r_0 \times p_{01} + r_1 \times p_{11} + r_2 \times p_{21} + r_3 \times p_{31} + r_4 \times p_{41}$$

$$s_2 = r_0 \times p_{02} + r_1 \times p_{12} + r_2 \times p_{22} + r_3 \times p_{32} + r_4 \times p_{42}$$

then:

$$s_0 = r_0 + r_1$$

$$s_1 = r_2 + r_3$$

$$s_2 = r_0 + r_2 + r_4$$

The L-syndrome circuit for this code is shown in Fig. 1.

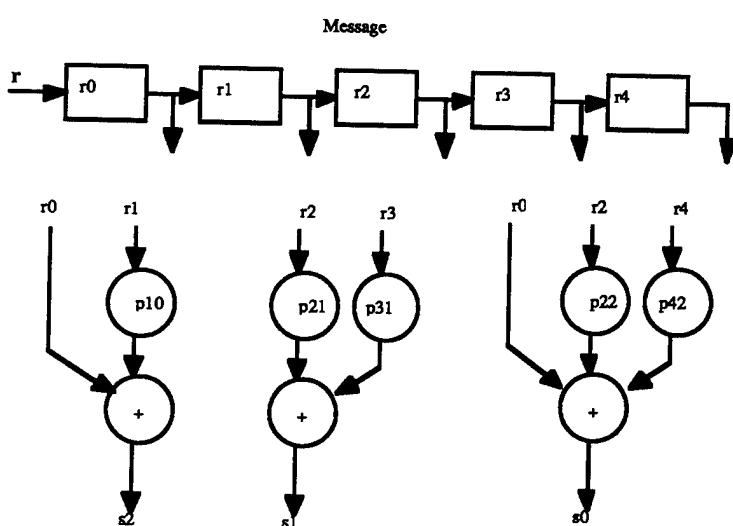


Fig. 57. L-syndrome circuit for the (5,2,3) Lexicographic code of Appendix Table 1.

Only codewords have  $g$  values equal to zero<sup>13</sup>. For all other  $z$  vectors of  $F_2^5$ ,  $g$  is the syndrome-B relative to the parity check matrix. The  $g$  values also define the cosets in  $F$ . For example, the definition of a coset<sup>14</sup> is:

If  $C$  is an  $[n, k]$  linear code over a field with  $q$  elements, then for any vector,  $z$ , the set

$$z + C = \{z + x : x \in C\}$$

is called a *coset* (or *translate*) of  $C$ . Every vector  $y$  is in some coset (in  $y + C$ , for example).  $z$  and  $y$  are in the same coset iff  $(z-y)$  exists in  $C$ . Each coset contains  $q^k$  vectors. For example, in the case of the  $n = 5$ ,  $k = 2$ ,  $d = 3$ , lexicographic code of Appendix, Table 2:

$g$  value of 0 - 4 cases (the codewords)  
 $g$  value of 1 - 4 cases  
 $g$  value of 2 - 4 cases  
 $g$  value of 3 - 4 cases  
 $g$  value of 4 - 4 cases  
 $g$  value of 5 - 4 cases  
 $g$  value of 6 - 4 cases  
 $g$  value of 7 - 4 cases

i.e.,  $q = 2$ ,  $k = 2$ ,  $q^k = 4$ .

Thus, the set  $F$  of all vectors,  $z$ , can be partitioned into cosets of  $C$ :

$$F^n = C \cup (z_1 + C) \cup (z_2 + C) \cup \dots \cup (z_t + C)$$

and  $t = g^{n-k} - 1$ . In the case of the lexicographic code of Table 1, with  $n=5$  and  $k=2$ ,  $t = 2^{5-2} - 1 = 7$ , i.e.,

$z_1 = \text{all } z \text{ vectors with } g = 1$   
 $z_2 = \text{all } z \text{ vectors with } g = 2$   
 $\dots$   
 $z_7 = \text{all } z \text{ vectors with } g = 7$ .

If a decoder receives the vector  $r$ , then  $r$  must belong to some coset in the above Eq., e.g.,  $r = z_i + x$  ( $x \in C$ ). The error vectors are:

$$e = r - x' = z_i + x - x' = z_i + x'' \in z_i + C$$

If a  $r$  vector is received, then a minimum weight vector,  $e$ , is chosen for the coset containing  $r$  and  $r$  is decoded as  $x = r - e$ . This minimum weight vector in a coset is called the *coset leader*.

The syndrome,  $s$ , can be expanded in terms of the transmitted codeword,  $v$ , and the error vector,  $e$ :

---

<sup>14</sup> MacWilliams, F.J. & Sloane, N.J.A., *The Theory of Error-Correcting Codes*, North-Holland, 1997, page 15.

$$s = r \cdot H^T = (v + e)H^T = v \cdot H^T + e \cdot H^T$$

As  $v \cdot H^T = 0$ ,

$$s = e \cdot H^T$$

Multiplying this out gives:

$$s_0 = e_0 + e_{n-k}p_{00} + e_{n-k+1}p_{10} + \dots + e_{n-1}p_{k-1,0}$$

$$s_1 = e_1 + e_{n-k}p_{01} + e_{n-k+1}p_{11} + \dots + e_{n-1}p_{k-1,1}$$

.

.

.

$$s_{n-k-1} = e_{n-k-1} + e_{n-k}p_{0,n-k-1} + e_{n-k+1}p_{1,n-k-1} + \dots + e_{n-1}p_{k-1,n-k-1}$$

with the syndrome digits being a linear combination of the error digits. These Eq.s also reveal that the  $n-k$  linear equations do not have a unique solution, but have  $2^k$  solutions. But this agrees with the observation, above, that each coset has  $q^k$  vectors, where  $q=2$ . Therefore, to determine the true error vector from a coset of  $2^k$  vectors, the most probable error pattern is chosen satisfying the above Eq.s.

The most probable error is understood as follows. Considering again the  $n = 5$ ,  $k = 2$ ,  $d = 3$ , lexicographic code of Appendix, Table 2, suppose the transmitted code word is  $v = (0,0,1,1,1)$  and the received code word is  $r = (0,0,1,0,1)$ . The L-syndrome for the received codeword is:

$$s = r \cdot H^T = (0 \ 1 \ 0),$$

and the error digits are related to the L-syndrome digits by:

$$0 = e_0 + e_1$$

$$1 = e_2 + e_3$$

$$0 = e_0 + e_2 + e_4$$

There are  $2^k$  or  $2^2 = 4$ , for  $k = 2$  error patterns that satisfy these equations. The possibilities are:

$$\begin{array}{ccccc} e_0 & e_1 & e_2 & e_3 & e_4 \\ \mathbf{e} = (0 & 0 & 0 & 1 & 0) \\ \mathbf{e} = (1 & 1 & 0 & 1 & 1) \\ \mathbf{e} = (0 & 0 & 1 & 0 & 1) \\ \mathbf{e} = (1 & 1 & 1 & 0 & 0) \end{array}$$

as the first possible error vector,  $e = (0 \ 0 \ 0 \ 1 \ 0)$ , has the smallest number of nonzero components, it is taken as the true error vector. Therefore the received vector,  $r = (0 \ 0 \ 1 \ 0$

1) is decoded into  $v^* = r+e = (0 \ 0 \ 1 \ 0 \ 1) + (0 \ 0 \ 0 \ 1 \ 0) = (0 \ 0 \ 0 \ 1 \ 1 \ 1)$ , which is the transmitted vector,  $v$ .

This error correction can be looked at from another viewpoint. An L-generator matrix for the  $n = 5$ ,  $k = 2$ ,  $d = 3$ , Lexicographic code is:

$$G = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The L-standard array is:

Standard Array for the Lexicographic Code (5,2)					
Basis	Coset Leader	00111	11001	11110	
1	$e_1 = 00000$	00110	11000	11111	
2	$e_2 = 000100$	00101	11011	11100	
3	$e_3 = 001100$	00011	11101	11010	
4	$e_4 = 010000$	01111	10001	10110	
5	$e_5 = 000010$	10111	01001	01110	
	$e_6 = 010100$	01101	10011	10100	
	$e_7 = 100010$	10101	01011	01100	

in which the basis vectors have been shaded.

This standard array has  $2^k$  disjoint columns, so  $k = 2$ . Each column consists of  $2^{(n-k)}$   $n$ -tuples. That is,  $2^{(5-2)} = 8$ , so  $8 \times 5$ -tuples. The coset leaders are  $e_2, e_3 \dots e_8$ . Each coset leader is of smallest weight than other members of the set. The significant result is that all row entries have identical  $g$ 's, e.g., the same standard array plotted in terms of the  $g$  values is<sup>15</sup>:

$g$  values of the  $z$  vectors of the Lexicographic Code (5,2)

0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7

<sup>15</sup> Proof of this empirical finding can be found in: Monroe, L., Binary Greedy Codes. *Congressus Numerantium* 104, 49-63, 1994.

This reflects the theorem that states that two vectors are in the same coset of  $C$  iff they have the same syndrome<sup>16</sup>. Supposing  $(0\ 0\ 1\ 0\ 1)$  is received, this is in the third row of the second column. The coset leader for this row is  $(0\ 0\ 0\ 1\ 0)$ , so the received signal is decoded correctly as  $(0\ 0\ 1\ 0\ 1) - (0\ 0\ 0\ 1\ 0) = (0\ 0\ 1\ 1\ 1)$  as before.

Compare the above analysis based on the Greedy-code method, with conventional methods. The S-parity check matrix for the Lexicographic (5,2) code (with the 4 codewords  $\{0, 0, 0, 0, 0\}$ ,  $\{0, 0, 1, 1, 1\}$ ,  $\{1, 1, 0, 0, 1\}$  and  $\{1, 1, 1, 1, 0\}$ ) is:

$$H = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$H^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

and the S-generator matrix is:

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

The S-generator matrix for the Lexicographic (5,2) code has the components:

$$s_0 = r_2 + r_3 + r_4$$

$$s_1 = r_1 + r_3 + r_4$$

$$s_2 = r_0 + r_4$$

Taking the previous example again, suppose  $(0\ 0\ 1\ 0\ 1)$  is received, when  $(0\ 0\ 0\ 1\ 1\ 1)$  was transmitted. Then  $s_0 = 1$ ,  $s_1 = 1$ ,  $s_2 = 0$ , and

$$1 = e_2 + e_3 + e_4$$

$$1 = e_1 + e_3 + e_4$$

$$0 = e_0 + e_4$$

There are  $2^k$  or  $2^2 = 4$ , for  $k = 2$  error patterns that satisfy these equations (but not the same four error patterns generated using L-generator matrix above). The possibilities this time are:

---

<sup>16</sup> MacWilliams, F.J. & Sloane, N.J.A., *The Theory of Error-Correcting Codes*, North-Holland, 1997, page 17, Theorem 4.

$$\begin{array}{ccccc}
 e_0 & e_1 & e_2 & e_3 & e_4 \\
 \mathbf{e} = (0 & 0 & 0 & 1 & 0) \\
 \mathbf{e} = (0 & 1 & 1 & 0 & 0) \\
 \mathbf{e} = (1 & 0 & 0 & 0 & 1) \\
 \mathbf{e} = (1 & 1 & 1 & 1 & 1)
 \end{array}$$

But the outcome is the same as before. As the first possible error vector,  $\mathbf{e} = (0 \ 0 \ 0 \ 1 \ 0)$  - which is the same error vector generated by the previous method - has the smallest number of nonzero components, it is taken as the true error vector. Therefore the received vector,  $\mathbf{r} = (0 \ 0 \ 1 \ 0 \ 1)$  is decoded into  $\mathbf{v}^* = \mathbf{r} + \mathbf{e} = (0 \ 0 \ 1 \ 0 \ 1) + (0 \ 0 \ 0 \ 1 \ 0) = (0 \ 0 \ 0 \ 1 \ 1)$ , which is, again, the transmitted vector,  $\mathbf{v}$ . The standard array is the same in both formulations.

The question arises: How is a message encoded in the Lexicographic formulation? The Lexicographic formulation can be related to the Systematic formulation as follows.

Commence with the Lexicographic parity check matrix as given above:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \text{row 1} \\ \text{row 2} \\ \text{row 3} \end{array}$$

Add rows 1 and 3, and replace row 3, giving:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \text{row 1} \\ \text{row 2} \\ \text{row 3} \end{array}$$

Add rows 3 and 2, replacing row 2, giving:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \text{row 1} \\ \text{row 2} \\ \text{row 3} \end{array}$$

Add rows 2 and 3, replacing row 3, giving:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad \begin{array}{l} \text{row 1} \\ \text{row 2} \\ \text{row 3} \end{array}$$

Add rows 1 and 2, replacing row 1, giving:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad \begin{array}{l} \text{row 1} \\ \text{row 2} \\ \text{row 3} \end{array}$$

The parity check matrix is now in the form of the S-parity check matrix shown above.

The same transformation can be performed with the L-generator matrix, which is:

$$G = \begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \begin{matrix} \text{row 1} \\ \text{row 2} \end{matrix}$$

Add rows 1 and 2, replacing row 1:

$$G = \begin{bmatrix} g_0 \\ g_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \begin{matrix} \text{row 1} \\ \text{row 2} \end{matrix}$$

The generator matrix is now in the form of the S-generator matrix shown above, and messages can be encoded in the normal way.

All three of the Lexicographic-Greedy (7,4) codes of Appendix Tables 7-9 have the same generator matrix:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

because all three code generations, based on different basis vectors, produce the same code words (i.e., the code vectors for which  $g = 0$ ). However, the plots, Tables 39-42 above, of the lexicographic ordering of these codes according to the different  $g$ -values, i.e., the ordering of the coset vectors or noncode words, indicate (1) symmetries underlying that ordering, and (2) differences between the three methods of code generation due to the differences in choice in basis vectors. The standard arrays for the three codes, Appendix Tables 4-6, also reveal that whereas in the case of the (7,4) code shown in Appendix Table 7 the coset leaders are also the basis vectors, this is not so for the codes shown in Appendix Tables 8 and 9, even although the codes themselves generated from all three sets of basis vectors are the same. This finding will have a bearing on the design of efficient error correction codes and will be investigated further.

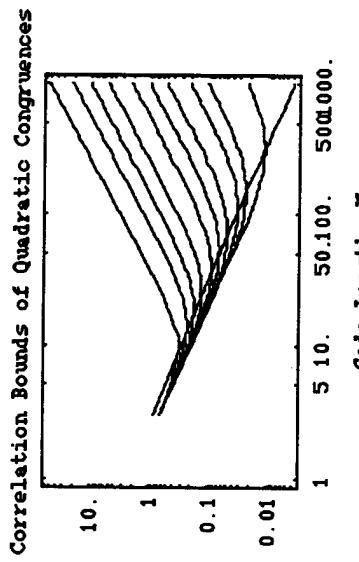


Fig. 58

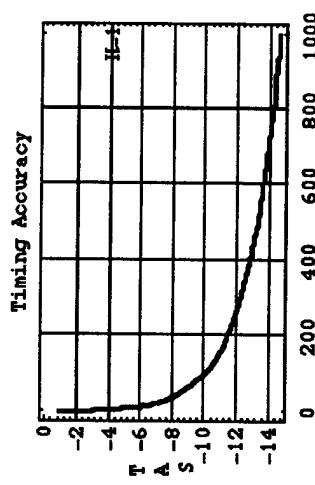


Fig. 60

Fig. 58. Time hopping and Frequency hopping: Cross-correlation bounds for extended quadratic congruence codes for  $BT$  or  $tT = 2,000, 5,000, 10,000, 20,000, 50,000, 100,000, 200,000, 500,000, 1,000,000$  and  $5,000,000$ . The loci of minimum values is also given. After<sup>17</sup>.

Fig. 59. Time hopping: Timing accuracy and stability (TAS) requirements versus the length of code  $N$  for  $H = 1, 10, 1000$ .

Fig. 60. Time hopping: Timing accuracy and stability requirements versus the length of code  $N$ , for  $H = 1$ , with clock setting every slot.

Fig. 61. Time hopping: Timing accuracy and stability requirements versus the length of code  $N$ , for  $H = 1$ , with clock setting every frame.

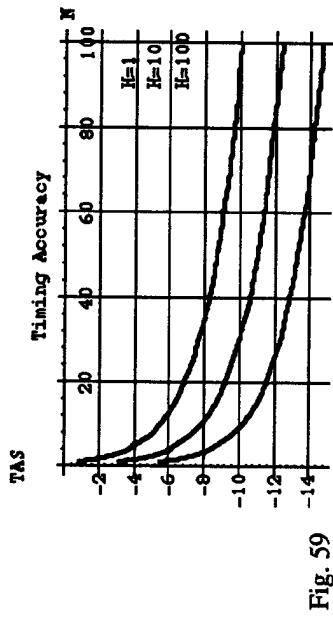


Fig. 59

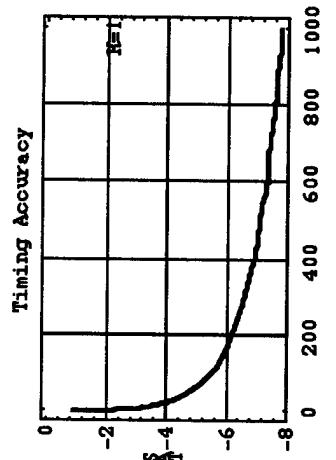


Fig. 61

<sup>17</sup>Bellegarda, J.R. & Titlebaum, E.I., Time-frequency hop codes based upon extended quadratic congruences. *IEEE Trans. Aerospace & Electronic Systems*, 24, 726-742, 1988.

## 1.2 Bit Error Analysis

Azizoglu et al<sup>18</sup> addressed performance limitations in optical CDMA scenarios comparable to those which may be predicted to occur in future OTDM networks. With reference to previous examples of codes based on the primary number 11, the following variables are defined:

- $F$  chips constitute, in our terminology, a superframe. For a  $11 \times 11$  code matrix,  $F = 11^2 = 121$ .
- Each of these chips is of duration  $\tau_c$  with  $T_b = F\tau_c$  being the duration of the superframe.  $K$  of the  $F$  chips are occupied. In the  $p = 11$  codes we have considered,  $K = 11$ .

The informational encoding which Azizoglu et al consider are the cases of:

- users in the network sending their data in an on-off modulation format, so that for a "0" bit a user sends nothing and for a "1" bit a temporal signature is sent.

Clearly, informational encoding could be:

- a pulse-position modulation scheme.

For optical orthogonal codes (OOC), two code words cannot overlap at more than one pulse position. As there are  $K^2$  ways of pairing the  $K$  pulses of two users, the probability that a pulse belonging to a particular user overlapping with one of the pulses of the desired user is:

$$q = \frac{K^2}{2F},$$

where the factor 1/2 is due to the probability that the interferer is transmitting a "1" - i.e., that the first form of informational encoding has been chosen.

If  $M$  is the number of users, and  $I$  the number of interfering users, the *pattern of interferences* is described by a an interference vector,  $\alpha(l)$ . Given that there are  $I$  interfering users - each interfering at exactly one pulse position - there is a variety of possible interference patterns. Therefore, the vector,  $\alpha(l)$ , is a  $K$ -dimensional vector whose  $i$ th element,  $\alpha_i(l)$ , represents the number of pulses that overlap with the  $i$ th pulse of the desired user. As *every interfering user contributes one and only one pulse*<sup>19</sup>, the vector satisfies:

$$\sum_{i=1}^K \alpha_i(l) = I \quad \alpha_i(l) \in \{0, 1, \dots, I\}.$$

If the receiver is *not* hard-limited, then:

<sup>18</sup> Azizoglu, M.Y., Salehi, J.A. & Li, Y., On the performance of fiber-optic CDMA systems. *Proc. IEEE GLOBECOM 1990*, San Diego, CA, Dec. 1990, pp. 1861-1865;  
Azizoglu, M.Y., Salehi, J.A. & Li, Y., Optical CDMA via temporal codes. *IEEE Trans. Comm.*, 40, 1162-1170, 1992.

<sup>19</sup> This assumption is questionable.

$$\sum_{i=1}^K \alpha_i \geq Th,$$

and an error will occur if  $l \geq Th$ . In the hard-limited case,  $Th = K$  and Azizoglu et al show that:

$$P_E = \frac{1}{2} \sum_{i=0}^K (-1)^i \binom{K}{i} \left(1 - \frac{qi}{K}\right)^{M-1},$$

for the hard-limiting case,  $Th = K$ . This probability of error is shown in Fig. 16 for three values of  $M$ , the number of users.

If the restrictions on the auto- and cross-correlations are relaxed from  $\lambda = 1$  to  $\lambda = 2$ , upper and lower bounds can be given to the error probability:

$$\text{lower bound : } P_E \geq \sum_{i=0}^K (-1)^i \binom{K}{i} \left[1 - \frac{pi}{K} \left(2 - \frac{i}{K}\right)\right]^{M-1},$$

$$\text{upper bound : } P_E \leq \sum_{i=0}^{K/2} (-1)^i \binom{K}{i} \left[1 - \frac{pi}{K} \left(2 - \frac{i}{K}\right)\right]^{M-1},$$

where  $p = K^2/4F$ . Upper and lower bounds are shown in Fig. 17, for  $M=50$ , and Fig. 18. for  $M=10$ .

Kostic and Titlebaum<sup>20</sup> have also addressed the problem of error for two sets of quadratic congruence codes defined with respect to the bound on the maximum value of the inner product between two arbitrary sequences being equal to  $2 - C'_{QC}$ , or equal to  $1 - C_{QC}$ . Interference is represented by a probability distribution function (pdf) of a random variable,  $I$ . For the set,  $C'_{QC}$  and prime  $p$ , the pdf is:

$$\begin{aligned} pdf_{C'_{QC}} &= \frac{1}{p+1} \delta(I_n - 0) + \frac{p}{p+1} \delta(I_n - 1), \text{ and} \\ E(I_n) &= 1, \quad \sigma^2(I_n) = \frac{p-1}{p}; \end{aligned}$$

and for the set,  $C_{QC}$ , the pdf is:

$$\begin{aligned} pdf_{C_{QC}} &= \frac{3p-1}{4p} \delta(I_n) + \frac{1}{2p} \delta(I_n - 1) + \frac{p-1}{4p} \delta(I_n - 2), \text{ and} \\ E(I_n) &= 1, \quad \sigma^2(I_n) = \frac{3p-2}{4p}. \end{aligned}$$

---

<sup>20</sup> Kostic, Z. & Titlebaum, E.L., The design and performance analysis for several new classes of codes for optical synchronous CDMA and for arbitrary-medium time-hopping synchronous CDMA communication systems. *IEEE Trans. Comm.*, 42, 2608-2617, 1994.

With the variables, *A*, *B* and *C*, defined as:

$$A = \frac{p-1}{2p}, B = \frac{1}{p}, C = \frac{p-1}{2p},$$

$$A = \frac{3p-1}{4p}, B = \frac{1}{2p}, C = \frac{p-1}{4p}$$

for the sets  $C'_{QC}$  and  $C_{QC}$ , respectively, the error probability is:

$$PE = \frac{1}{2} \left\{ \sum_{i=Th}^{N-2} \sum_{j=\left[\frac{i+1}{2}\right]}^i A^{N-1-j} B^{2j-i} C^{i-j} \frac{(N-1)!}{(2j-i)!(i-j)!(N-1-j)!} + \right. \\ \left. \sum_{i=N-1}^{2(N-1)} \sum_{j=\left[\frac{i+1}{2}\right]}^{N-1} A^{N-1-j} B^{2j-i} C^{i-j} \frac{(N-1)!}{(2j-i)!(i-j)!(N-1-j)!} \right\}$$

Fig.s 19-22 show the (log) error probability versus matched filter threshold for several codes and users. Fig.s 23-26 show the ratio of the error probability to the number of sequences in a set versus the matched filter threshold for a number of codes and users.

These calculations indicate that the probability of error decreases with (1) an increase of the matched filter threshold and (2) an increase in code length.

The upper bound of the probability of error for the case of chip synchronous is<sup>21</sup>:

$$PE = \frac{1}{2} \sum_{i=Th}^{N-1} \binom{N-1}{i} \left( \frac{K^2}{2F} \right)^i \left( 1 - \frac{K^2}{2F} \right)^{N-1-i},$$

where "Th" is a correlation receiver threshold,  $0 \leq Th \leq K$ . Fig. 27 indicates (a) as expected, a reduction of the number of chips per frame results in a lowering of performance; and (b) as the receiver threshold is raised, the performance improves.

Fig. 28 on the other hand, indicates that using more pulses in a sequence *does not result in a reduction in performance if one can raise the receiver threshold*. In fact, raising the receiver threshold due to the availability of more pulses in the code *results in a much greater gain in performance than any reduction in performance due to there being more pulses in the code*. However, if the number of pulses in the code were increased and the threshold were kept constant, then the performance *would* degrade.

Fig. 29 shows the result for the case of the number of slots available in a superframe, *F*, being much larger than the number of users, *N*. In this case, increasing the number of users results in a reduction of performance (increasing *N* from 10 to 50 is shown). This reduction in performance can be offset by hardlimiting the receiver. An optical hardlimiter, or threshold element, is defined as:

<sup>21</sup> Salehi, J.A. & Brackett, C.A., Code division multiple-access techniques in optical fiber networks - Part II: Systems performance analysis. *IEEE Trans. Comm.*, 37, 834-842, 1989.

$$g(x) = \begin{cases} 1, & x \geq 1 \\ 0, & 0 \leq x < 1 \end{cases}$$

That is, if an optical light intensity ( $x$ ) is greater than, or equal to, one, then hardlimiting gives an output of one; and if the optical light intensity is smaller than one, then the response of the hardlimiter is zero.

If *hard-limiting* is used on a receiver front-end, then the probability of error for the chip synchronous case is:

$$PE \leq \frac{1}{2} \left( \frac{K}{Th} \right) \prod_{m=0}^{Th-1} (1 - q^{N-1-m}); \quad q = 1 - \frac{K}{2F}; \quad p = 1 - q.$$

Fig. 30 (when compared with Fig. 28) shows that hard limiting increases performance by approximately 1.5 orders of magnitude. Fig.s 31 and 32 (when compared with Fig.s 27 and 29 indicate the same level of performance improvement.

In the case of asynchronous transmission with hard limiting, the probability of error is<sup>21</sup>:

$$PE \leq \frac{1}{2} \left( \frac{K}{Th} \right) \prod_{m=0}^{Th-1} \left[ 1 - \left\{ q^{N-1-m} + (N-1-m)pq^{N-2-m} + \sum_{i=2}^{N-1-m} \binom{N-1-m}{i} p^i q^{N-1-i-m} \left( 1 - Q\left(\frac{1-i/2}{\sqrt{i/12}}\right) \right) \right\} \right]$$

where

$$\begin{aligned} p &= \frac{K}{F}, \\ q &= 1 - p, \\ Q(x) &= \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left[-\frac{\mu^2}{2}\right] d\mu. \end{aligned}$$

Fig. 33 indicates the improvement of hard-limiting to the asynchronous case. The improvement is about 4-5 orders of magnitude.

Because the number of users is unacceptably low with the criterion  $\lambda = 1$ , the consequences of raising this criterion to  $\lambda = 2$  has been examined. With  $l_1$  users interfering at 1 position and  $l_2$  users interfering at 2 pulse positions, (giving the total number of interfering pulses =  $l_1 + 2l_2$ ), the error probability is:

$$P_E = \frac{1}{2} \sum_{\substack{l_1-2l_2 \geq Th \\ l_1-l_2 < M}} \Pr(l_1 l_2)$$

with

$$l_1 + l_2 < M,$$

$$M \leq \frac{(F-1)(F-2)}{K(K-1)(K-2)}.$$

The probability of error is then:

$$P_E = \frac{1}{2} - \frac{1}{2} \sum_{l_1=0}^{Th-1} \sum_{l_2=0}^{\lfloor (Th-1-l_1)/2 \rfloor} \frac{(M-1)!}{l_1! l_2! (M-1-l_1-l_2)!} q_1^{l_1} q_2^{l_2} (1-q_1-q_2)^{M-1-l_1-l_2}.$$

Azizoglu et al<sup>22</sup> have shown that the statistic:

$$c \approx \frac{q_1}{K^2/2F}$$

indicates the deviation of a code from the strict orthogonal optical code (OOC) criterion. For  $c = 1$  the code is an OOC. For  $c = 0$ , the code will have 2 overlaps in auto- and cross-correlation. Fig.s 34 and 35 show the error probability as a function of  $c$  for the optimum threshold:  $Th = K$ , with  $F = 1000$  and for the hard-limiting case. These Fig.s show that the error probability increases with decreasing  $c$ , *but the number of possible users increases*.

On introducing hard-limiting to the receiver, and with  $\lambda = 2$ , the lower bound on the error probability is:

$$P_E \geq \frac{1}{2} \sum_{i=0}^K (-1)^i \binom{K}{i} \left[ 1 - \frac{pi}{K} \left( 2 - \frac{i}{K} \right) \right]^{M-1},$$

and the upper bound is:

$$P_E \leq \frac{1}{2} \sum_{i=0}^{K/2} (-1)^i \binom{K/2}{i} \left( 1 - \frac{2pi}{K} \right)^{M-1}.$$

The upper and lower bounds are shown in Fig. 36, for  $M = 50$ ,  $F = 1000$ , and in Fig. 37, for  $M = 10$ ,  $F = 1000$ . A comparison with Fig. 38, for the  $\lambda = 1$  case, indicates that the performance with  $\lambda = 2$  is a few orders of magnitude poorer, but as pointed out by Azizoglu et al, more chips,  $K$ , can be used for the codes, thus giving compensatory performance - as Fig. 28 clearly shows. Therefore the conclusion must be that  $\lambda = 2$  codes may be as good in performance as  $\lambda = 1$  codes, while possibly offering the availability of more codes to more users.

---

<sup>22</sup> Azizoglu, M., Salehi, J.A. & Li, Y., Optical CDMA via temporal codes. *IEEE Trans. Comm.*, 40, 1162-1169, 1992.

## 1.4 Code Design

### 1.4.1 Perfect Error-Correcting Codes

The Greedy code algorithm, which generates codes according to a designed distance,  $d$ , at set  $n$ , provides superior codes than those generated by the conventional methods of commencing with a designed  $k$ , at set  $n$ . Greedy code algorithms provide *perfect* codes. The generation of perfect codes with designed distance  $d$ , which are *usable* means that the rate  $k/n$  must still be less than the channel capacity,  $C$ . Pictorial representation shows that the Table II (7,4) code of distance 3 is the most symmetric. However, this code was not considered optimum because the code rate,  $k/n$ , exceeds the channel capacity of 0.408. On the other hand, the code bounds derived are intended for *large*  $n$ . Therefore, further investigation could reveal that symmetry does play a role in choosing the optimum Lexicographic code. This task must be left for future work as generation of Lexicographic codes of large  $n$  by the Greedy code algorithm, although now straightforward because we have generated the code, requires lengthy code modification and data treatment.

In the following Tables and Figures we show that the generation of good error-correcting codes by the single criterion of high bit error correction,  $t$ , is insufficient to obtain good and *usable* codes. Other criteria, such as a code rate,  $k/n$ , permitted by the channel capacity and a minimization of the probability of error, are also determining factors.

For example, Table 4 shows various  $d$ ,  $k$ ,  $t$  and  $M$  values for an  $n = 7$  Linear Lexicographic Code generated with the Greedy algorithm. The highest  $t$  obtained is 3 and it might be supposed that the (7,1) or the (7,0) codes are equally usable. However, Table 5 shows that by the criterion of the lowest probability of error, only the (7,1) code is usable, but even that conclusion carries the penalty that only 2 (7,1) codes can be generated ( $M = 2$ ) at the designed distance of 7. This code is shown in Table 15 of the Appendix.

Table 4.

Designed distance,  $d$ , for  $n = 7$  Lexicographic Greedy perfect codes of Appendix: Tables 10-16 and the resulting tradeoffs with limits on bit error correction,  $t$ , with the number of message elements,  $k$ , and with the number of codes,  $M$ .

$d$	$n$	$k$	$t$	$M$
1	7	7	0	128
2	7	6	0	64
3	7	4	1	16
4	7	3	1	8
5	7	1	2	2
6	7	1	2	2
7	7	1	3	2
8	7	0	3	1

Table 5.

Probability of Error for  $n = 7$  Lexicographic Greedy perfect codes of Appendix Tables 10-16. Those with code rate,  $k/n$ , less than the Channel Capacity,  $C = 0.408$ , are highlighted. Of these, the row of the code with the smallest probability of error (0.010) is shaded. This code is found in Table 15 of the Appendix.

$d$	$n$	$k$	$t$	$Prob(\text{Error})$	$k/n$
1	7	7	0	0.660	1
2	7	6	0 I	0.660	0.857
3	7	4	1 II	0.264	0.571
4	7	3	1 III	0.264	0.429
5	7	1	2 IV	0.065	0.143
6	7	1	2 V	0.065	0.143
7	7	1	3 VI	0.010	0.143
8	7	0	3 VII	0.010	0

Figure 62 displays the same data for all codes in another way. The Figure shows that a high  $d$  implies a low  $k$  (hence a low  $M$ ), achieving a low probability of error, which is good, but at the price of a low code rate,  $k/n$ , which is not good.  $t$  approximately follows the direction of  $d$ .

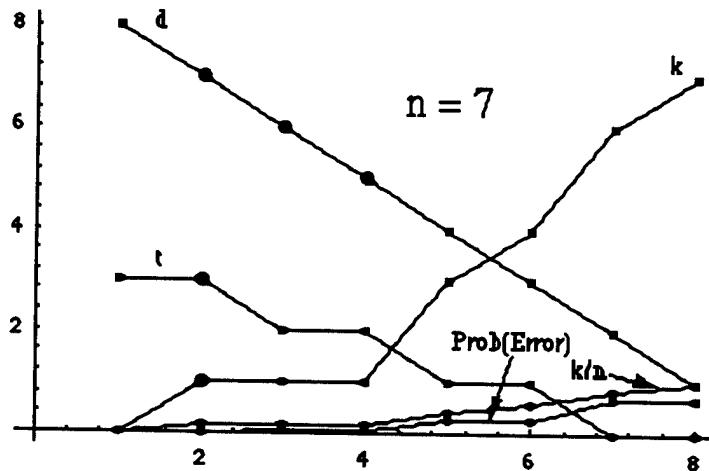


Fig. 62. Pictorial representation of Table 1. The data points for which  $k/n < C$ , are the 2nd, 3rd and 4th from the left (larger dots). Of these, the data point with the smallest probability of error (0.010) is the 2nd.

The situation improves with larger  $n$ . Fig. 63 shows the probability of error versus the probability of bit error at various  $t$  values for  $n = 7$ . Comparing this Figure with Fig. 64, which is for  $n = 1021$ , we see that a larger  $n$  supports a larger  $t$ , but with higher probability of bit errors at constant  $t$ . This point is made clearer in the Log-Log plots of Fig.s 66 and 67. These Fig.s show the advantage of a larger  $n$ . Although at the same  $t$  for both code lengths a smaller probability of bit error results in a larger word error at larger  $n$ , the larger  $n$  can support higher  $t$ 's and hence a smaller probability of word error is achieved.

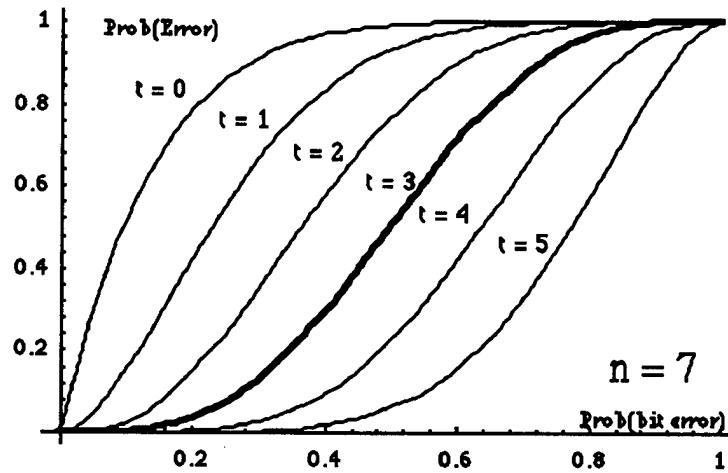


Fig. 63. The probability of error versus the bit error probability (probability of symbol error) for  $n = 7$ .  $1/k P_{\text{err}} \leq P_{\text{symb}} \leq P_{\text{err}}$ .  $C = 0.408$  and  $k/n < C$ , for  $t \geq 3$ .  $t = 4$  and  $5$  are unusable.

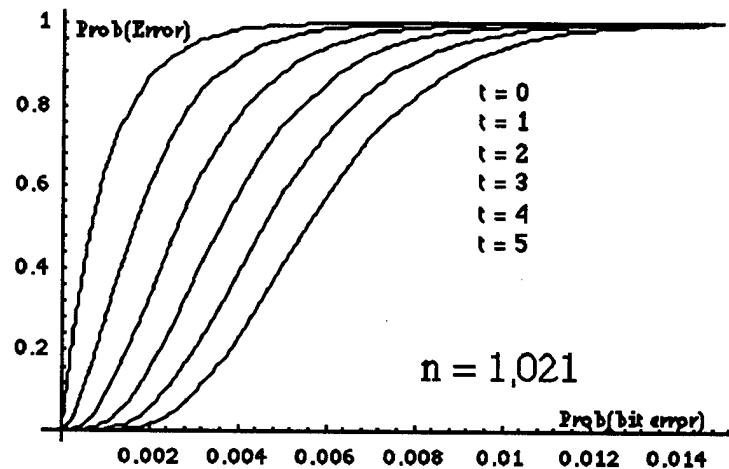


Fig. 64. The probability of error versus the bit error probability (probability of symbol error) for  $n = 1,021$ .  $1/k P_{\text{err}} \leq P_{\text{symb}} \leq P_{\text{err}}$ .  $C = 0.989$  and  $k/n < C$ , for all  $t$ 's shown. All  $t$ 's are usable, as are larger  $t$ 's not shown.

The impact of the channel capacity in the choice of codes is clearly shown in Fig. 65. The channel capacity,  $C$ , is 0.408 and only  $k = 0$  and 1 provides a rate,  $k/n$ , which is below that capacity. A final determination of the usable code is in terms of the probability of error (Table 5).

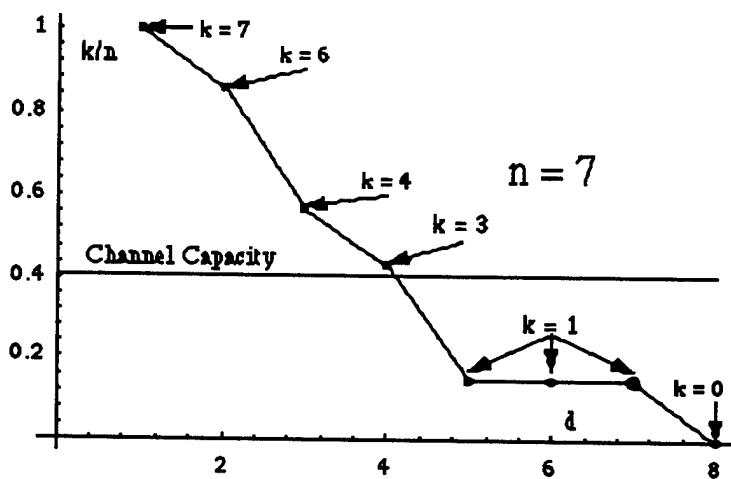


Fig. 65. Code rate  $k/n$  versus  $d$ , with  $k$  values represented. The only usable  $k/n$  rates less than  $C$  are for  $k = 1$ . Of these, the highlighted data point is optimum with respect to probability of error.

The essential features of a "good" error-correcting code are:

- that its distance,  $d$ , is as large as possible, so that  $\leq 2t$  corruptions can be detected, and  $\leq t$  bits of a codeword can be corrected, if  $d = 2t + 1$ . (If  $d = 2t$ , then  $2t - 1$  errors can be detected and  $t - 1$  errors corrected).
- that the code is constructed so that error detection and correction can be performed without the need to compare the received vector with all valid codewords.
- that the code is constructed with sufficient distance, but with also sufficient number of codewords,  $M = 2^k$ , generated.

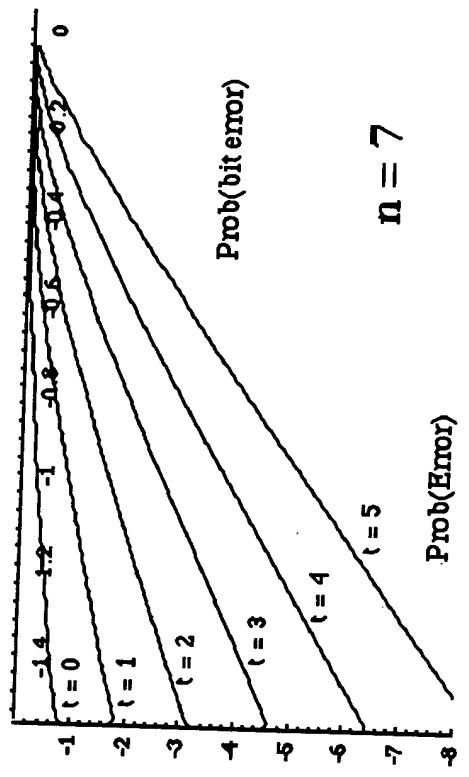


Fig. 66. Log-Log plot: Reducing the probability of bit error *and* increasing  $d$  (thereby increasing  $t$ ) results in a reduction in the word error. The probability of bit error is influenced by the size of  $n$ . The relation of  $t$  to  $n$  is shown in Fig. 68.

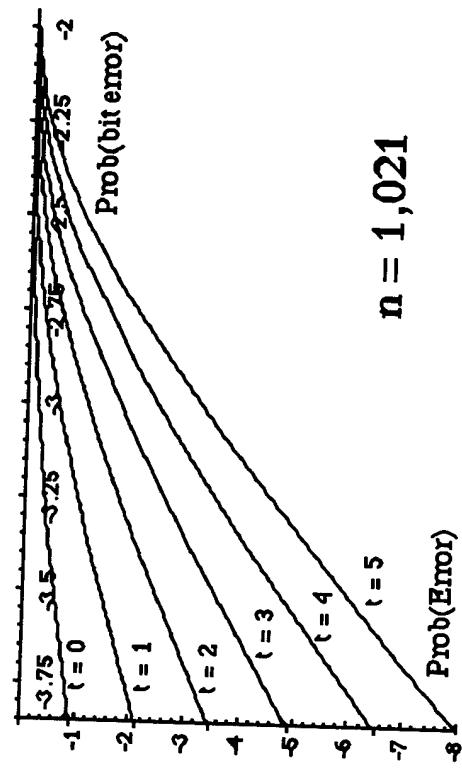


Fig. 67. Log-Log plot: Reducing the probability of bit error *and* increasing  $d$  (thereby increasing  $t$ ) results in a reduction in the word error. The probability of bit error is influenced by the size of  $n$ . The relation of  $t$  to  $n$  is shown in Fig. 68. Comparing this Fig. 66 with Fig. 67 (for  $n = 7$ ) shows that codes of large  $n$  can support much larger  $t$ .

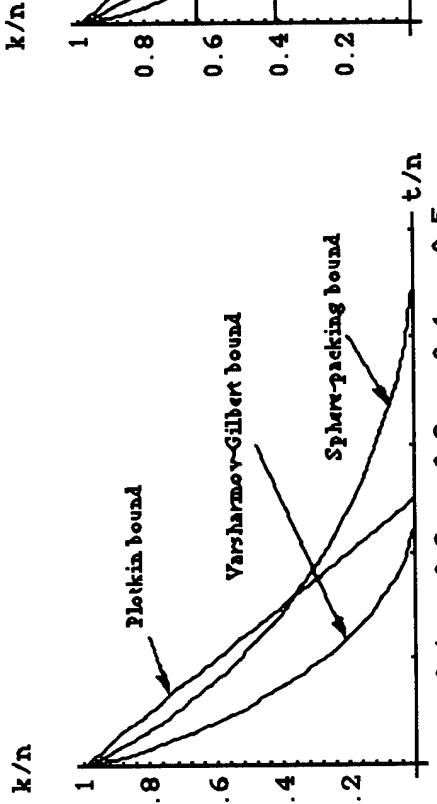


Fig. 68. Bounds on the coderate,  $k/n$ , and on  $t/n$ .

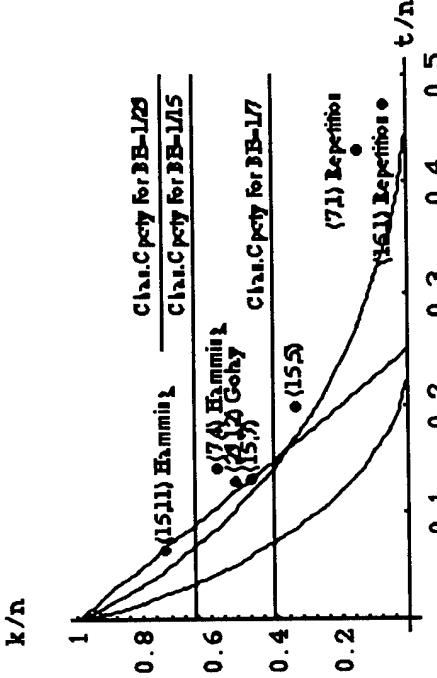


Fig. 69. Bounds on the coderate,  $k/n$ , and on  $t/n$  with codes  $(n, k)$  and channel capacities indicated. Only the rates for codes, (7,1), (15,1), (15,5), (15,7) and (23,12) are acceptable.

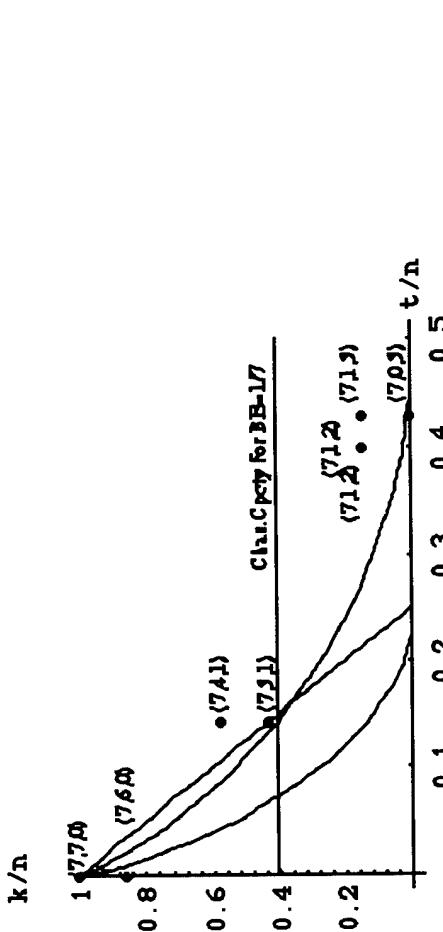


Fig. 70. Bounds on the coderate,  $k/n$ , and on  $t/n$  with  $(7, k, i)$  codes and channel capacity indicated. Only the rates for the codes (7,1,2), (7,1,3) and (7,0,3) are acceptable. Of these, only the (7,1,3) code is optimum with respect to probability of error (see Table 2, above).

These requirements dictate that each codeword be surrounded by a number of vectors,  $S(n, t)$ , which describe the codeword corrupted by single-bit errors, two-bit errors, etc., up to  $t$ -bit errors. This number is defined<sup>23</sup>:

$$S(n, t) = \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{t},$$

and

$$M(1 + S(n, t)) \leq 2^n.$$

For large  $n$ , the sphere-packing bound on  $M$  is:

$$\log_2 M \leq n(1 - H(\alpha)),$$

where

$$H(\alpha) = -(\alpha \log_2 \alpha + (1 - \alpha) \log_2 (1 - \alpha)) \quad (\text{the entropy function}),$$

$$\alpha = \frac{t}{n} = \frac{(d-1)}{2n}.$$

The code rate,  $k/n$ , of a code with  $k$  information bits is then defined as:

$$k/n \leq 1 - H(\alpha).$$

The major point, however, is that there is an upper limit on the number of codewords that can exist if a minimum distance,  $d$ , is a goal.

Shannon's theorem<sup>24</sup> states that:

$P_E = \text{Prob}(\text{Decoding Error}) = \text{Prob}(\text{No codeword}) + \text{Prob}(\text{Two or more codewords})$ , where

$$\text{Prob}(\text{Two or more codewords}) = (M - 1)(1/2^2) \left( \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{qn} \right),$$

$$q = (p + \varepsilon) = \alpha = t/n.$$

The above reduces to:

$$\text{Prob}(\text{Two or more codewords}) < \frac{M 2^{nH(q)}}{2^n},$$

$$M < 2^{2n(1-H(p))}, \text{ where}$$

$C = 1 - H(p)$  is the capacity of a binary symmetric channel.

---

<sup>23</sup> cf. Purser, M., *Introduction to Error-Correcting Codes*, Artech, Boston, 1995.

<sup>24</sup> Shannon, C.E., Communication in the presence of noise, *IRE Proc.*, 37, 1949.

With  $2^{n-k}$  possible syndromes and  $\binom{n}{w}$  error patterns of weight  $w$ , then for  $t$ -error correction, the following bound must be true:

$$1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t} \leq 2^{n-k}$$

This is the so-called *sphere-packing bound*. Another bound can be constructed by requiring linear independence of the columns of the parity check matrix,  $\mathbf{H}$ . This choice of columns results in:

$$\binom{n-1}{1} + \binom{n-1}{2} + \dots + \binom{n-1}{2t-1} < 2^{n-k} - 1.$$

This is known as the *Varshamov-Gilbert bound*.

Another bound, the *Plotkin bound*, is obtained by the observation that for given  $t$ , as  $n$  increases by 1 bit, the number of codewords at most doubles. The Plotkin bound for linear codes is:

$$\text{coderate} = k/n \leq 1 - 4t/n,$$

the Varshamov-Gilbert bound is:

$$\text{coderate} = k/n < 1 - H(2t/n)$$

and the sphere-packing bound is:

$$\text{coderate} = k/n < 1 - H(t/n)$$

Combining the Varshamov-Gilbert bound and the sphere-packing bound gives:

$$1 - H(2t/n) < k/n < 1 - H(t/n).$$

Fig. 68, above, shows these three bounds on coderate,  $k/n$ , and on  $t/n$  (for large  $n$ ). Fig. 69 plots values for a variety of codes and Fig. 70 plots values for  $n = 7$  codes. It can be seen that none of these really satisfy the bounds, but  $n$  is small for all the codes plotted. Shannon's theorem indicates that increasing  $n$  will result in a better result, but this will require practical devices providing long word lengths and such devices are not yet available.

In the light of the above, the following Linear Systematic (5,2) Code is examined:

$$\mathbf{G} = \left[ \begin{array}{cc|cc} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{array} \right] \quad \mathbf{H} = \left[ \begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{array} \right],$$

and compared with Linear Lexicographic (5,2) Codes. The standard array for the Linear Systematic code is:

Table 6: Linear Systematic (5,2) Code, $d = 3$				
$c_0$	$c_1$	$c_2$	$c_3$	syndrome
00000	10110	01011	11101	000
10000	<b>00110</b>	11011	01101	110
01000	11110	<b>00011</b>	10101	011
00100	<b>10010</b>	01111	11001	100
00010	<b>10100</b>	<b>01001</b>	11111	010
00001	10111	<b>01010</b>	11100	001
00000	01010	10101	01110	101
00001	00111	10110	01100	111

The standard array for one Linear Lexicographic code generated with the ordered basis set shown in column,  $c_0$ , is:

Table 7A: Linear Lexicographic (5, 2) Code, $d = 3$				
$c_0$	$c_1$	$c_2$	$c_3$	syndrome
00000	00111	11001	11110	000
00001	<b>00110</b>	<b>11000</b>	11111	001
00010	<b>00101</b>	11011	11100	010
00100	<b>00011</b>	11101	11010	100
01000	01111	<b>10001</b>	10110	110
10000	10111	<b>01001</b>	01110	111
00000	01010	10101	01110	101
00001	00111	10110	01100	011

In order to make a direct comparison with the Linear Systematic (5,2) code of Table 6, a second Linear Lexicographic (5,2) code is shown in Table 7B with the first six rows of the  $c_0$  column identical to Table 6. The code of Table 7B is completely described in Table 1 of the Appendix.

Table 7B: Linear Lexicographic (5, 2) Code, $d = 3$				
$c_0$	$c_1$	$c_2$	$c_3$	syndrome
00000	11100	10011	01111	000
10000	<b>01100</b>	<b>00011</b>	11111	100
01000	<b>10100</b>	11011	00111	010
00100	<b>11000</b>	10111	01011	001
00010	11110	<b>10001</b>	01101	011
00001	11101	<b>10010</b>	01110	111
01010	10110	11001	00101	161
01001	10101	11010	00110	110

In Tables 6, 7A and 7B the vectors of weight 2 are in bold. For all 2-error patterns above the double line there is unambiguous correspondence with the coset leader,  $c_0$ , and the syndrome. However, below the double line there is ambiguity concerning the coset leader assignment. Both the Systematic and the Lexicographic codes have this ambiguity.

The ambiguity of the assignment of the coset leaders of the final two rows is also depicted in the following Tables 8, 9A and 9B, showing the weights of the three codes.

Table 8: Linear (5,2) Systematic Code: Weights				
$c_0$	$c_1$	$c_2$	$c_3$	syndrome
0	3	3	4	000
1	2	4	3	110
1	4	2	3	011
1	2	4	3	100
1	2	2	5	010
1	4	2	3	001
2	3	3	2	101
2	3	3	2	111

Table 9A: Linear (5,2) Lexicographic Code: Weights				
$c_0$	$c_1$	$c_2$	$c_3$	syndrome
0	3	3	4	000
1	2	2	5	001
1	2	4	3	010
1	2	4	3	100
1	4	2	3	110
1	4	2	3	111
2	3	3	2	101
2	3	3	2	111

Table 9B: Linear (5,2) Lexicographic Code: Weights

$c_0$	$c_1$	$c_2$	$c_3$	syndrome
0	3	3	4	000
1	2	2	5	100
1	2	4	3	010
1	2	4	3	001
1	4	2	3	011
1	4	2	3	111
2	5	3	2	101
2	3	3	2	110

Tables 10, 11A and 11B show the  $g$  values of the Linear Lexicographic (5,2) codes (Tables 11A and 11B), assigned to the Linear Systematic (5,2) code (Table 4):

Table 10:  $g$  values of the Linear Lexicographic (5,2) Codes (Tables 11A and 11B) corresponding to the Linear Systematic (5,2) Code of Table 4.  
 \* Codewords of the Linear Lexicographic Code of Table 11A, below.

$c_0$	$c_1$	$c_2$	$c_3$	$s$
*	4	7	3	000
5	1	2	6	110
4	*	3	7	011
3	7	4	*	100
2	6	5	1	010
1	5	6	2	001
1	5	6	2	101
4	*	3	7	111

Table 11A:  $g$  values of the Lexicographic (5,2) Code.

$c_0$	$c_1$	$c_2$	$c_3$	$s$
0	0	0	0	000
1	1	1	1	001
2	2	2	2	010
3	3	3	3	100
4	4	4	4	110
5	5	5	5	111
6	6	6	6	101
7	7	7	7	011

Table 11B:  $g$  values of the Lexicographic (5,2) Code.

$c_0$	$c_1$	$c_2$	$c_3$	$s$
0	0	0	0	000
1	1	1	1	001
2	2	2	2	010
3	3	3	3	100
4	4	4	4	110
5	5	5	5	111
6	6	6	6	101
7	7	7	7	011

The major difference shown by these Tables between the Linear Systematic (5,2) and the Linear Lexicographic (5,2) codes is that the Linear Lexicographic code is a perfect code, the Linear Systematic code is not. A perfect code is defined as follows.

- There are  $2^{n-k}$  possible distinct syndromes and  $\binom{n}{w}$  distinct error patterns of weight  $w$ , therefore, for  $t$  error correction, we require that:

$$1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t} \leq 2^{n-k}.$$

For a given  $(n - k)$  and  $n$ , this gives an upper bound for the number of correctable bits in error,  $t$ . An  $(n, k)$  linear code for which the inequality becomes an equality is a *perfect code*.

- A perfect  $t$ -error correcting code can correct all errors of weight  $\leq t$ .
- In the case of a perfect code, the spaces of radius  $t$  around a codeword are disjoint and together contain *all* vectors of length  $n$ .
- A nontrivial perfect code over any field must have the same parameters  $n, M$  and  $d$  as one of the Hamming and Golay codes<sup>25</sup>.
- Levenshtein proved the linearity of the Lexicographic Codes and that the Hamming codes are Lexicographic Codes<sup>26</sup>. In particular, the Lexicographic Codes for  $n = 2^m - 1$  and  $d = 3$  are the Hamming Codes.
- The Lexicographic Code for  $n = 23$  and  $d = 7$  is the binary Golay Code. The Lexicographic Code for  $n = 24$  and  $d = 8$  is also the binary Golay Code<sup>27</sup>.

<sup>25</sup> Tietäväinen, A., On the nonexistence of perfect codes over finite fields. *SIAM J. Appl. Math.*, 24, 88-96, 1973;

Van Lint, J.H., *Introduction to Coding Theory*, 2nd Edition, Springer, New York, 1992, page 102; MacWilliams, F.J. & Sloane, N.J.A., *The Theory of Error-Correcting Codes*, North Holland, 1977, page 180.

<sup>26</sup> Levenshtein, V.I., A class of systematic codes. *Dokl. Akad. Nauk*, 1, 368-371, 1960.

<sup>27</sup> Bruandi, R.A. & Pless, V.S., Greedy codes. *J. Combinatorial Theory, A* 64, 1993, 10-30.

Perfect codes permit the sphere-packing bound to be exactly obtained. Therefore their use is preferred. The Lexicographic codes, as perfect codes, also permit a designed distance,  $d$ . Therefore, there is an importance to the mathematical structure underlying the Lexicographic Codes. Figures 71-74 and 75-81 provide pictorial representations of the  $g$ -values or lexicographic ordering of the  $n = 7$  codes. Fig.s 71 and 76 show that the Appendix Table 11 (7,4) code of distance 3 is the most symmetric. However, this code was not considered optimum because the code rate,  $k/n$ , exceeds the channel capacity of 0.408. On the other hand, the code bounds derived are intended for *large n*. Therefore, further investigation *could* reveal that symmetry does play a role in choosing the optimum Lexicographic code. This task must be left for future work as generation of Lexicographic codes of large  $n$  by the Greedy code algorithm, although now straightforward because we have written the code, requires lengthy code modification and data treatment.

#### 1.4.2 The Trade-off between Low Out-of-Phase Auto-Correlation and Low Cross-Correlation in Orthogonal Codes.

Techniques have long been known for generating hopping patterns which have low cross-correlation. In particular, Reed-Solomon codes<sup>28</sup> can be divided into disjoint sets, each of which contains all the cyclic time-frequency translates of any member of the subset. By selecting one member from each subset, a family of patterns is generated with low cross-correlation<sup>29</sup>. The Mersereau and Seay selection technique originally addressed time and frequency shifts in a frequency hopping scheme. Here, we adapt it to address a time hopping scheme. The selection technique is to arrange the R-S codewords into sets and choose each code word by not permitting cyclic frame or cyclic slot shifts. In order to achieve these conditions, the following rules apply:

1. Let a codeword  $c_i$  be defined as:  $c_i = [n_0, n_1, \dots, n_k]G$ . A value for the first coordinate  $n_0$  is chosen to be common to all  $(k+1)$ -tuples.
2. A nonzero value for the second coordinate  $n_1$  is chosen to be common to all  $(k+1)$ -tuples.
3. All possible combinations of values are assigned to the remaining coordinates.

These conditions imply that if  $(n_0, n_1, \dots, n_k)$  is an allowed  $(k+1)$ -tuple, then  $(n_0, n_1\alpha^t, \dots, n_k\alpha^{kt})$  for all  $t$  are not allowed  $(k+1)$ -tuples.

These conditions also imply that  $n_1 \neq 0$ , which means that the cyclic time shifts of a given codeword are distinct. (If  $n_1 = 0$ , cyclic time shifts of the original code word can exist giving the original code word, i.e., the period of the waveform is less than  $p-1$  (the number of slots per frame)). However, if there is a codeword corresponding to  $n_1 \neq 0$ , then if  $c_2 = [n_0, n_1\alpha^t, \dots, n_k\alpha^{kt}]G$  is the cyclic time shift of  $c_i$  with time shift  $t$ , it follows that no time shift of  $c_i$  is equal to  $c_2$ . This follows by choosing one codeword from each subset having  $n_1 \neq 0$ . Thus the Merseau-Seay selection procedure provides having at most  $k$  overlaps for all time shifts.

<sup>28</sup>Reed, I.S. & Solomon, G., Polynomial codes over certain finite fields. *J. of the Society for Industrial and Applied Mathematics (SIAM)*, 8, 300-304, 1960.

<sup>29</sup>Mersereau, R.M. & Seay, T.S., Multiple access frequency hopping patterns with low ambiguity. *IEEE Trans. Aerospace & Electronic Systems*, AES-17, 571-578, 1981.

The following examples (Fig.s 53-56), adapted for time hopping (instead of frequency hopping) test this prediction.

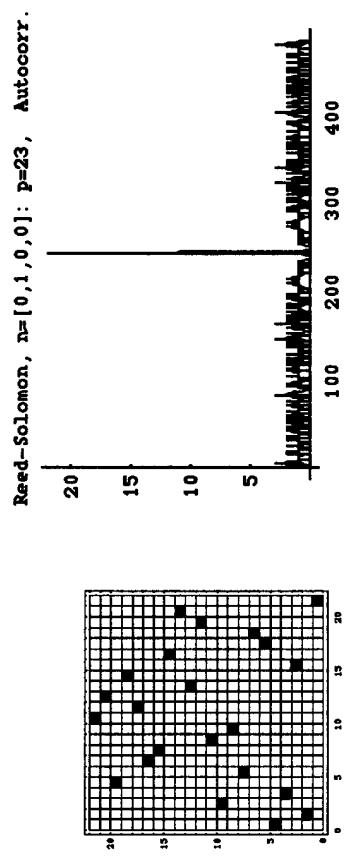


Fig. 53

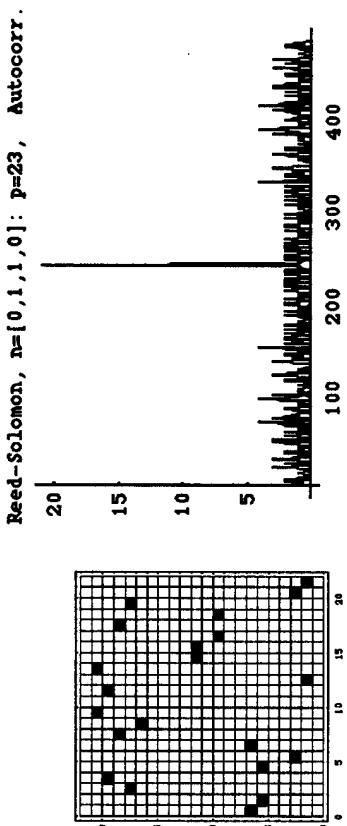


Fig. 53

$$G = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 5 & 5^2 & 5^3 & \dots & 5^{22} = 1 \\ 5^2 & 5^4 & 5^6 & \dots & 1 \\ 5^3 & 5^6 & 5^9 & \dots & 1 \end{bmatrix}, n=[0,1,0,0], p=23.$$

Fig. 53. Reed-Solomon code:  $c = nG$ ,  $G =$ 

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 5 & 5^2 & 5^3 & \dots & 5^{22} = 1 \\ 5^2 & 5^4 & 5^6 & \dots & 1 \\ 5^3 & 5^6 & 5^9 & \dots & 1 \end{bmatrix}, n=[0,1,1,0], p=23.$$

Fig. 54. Reed-Solomon code:  $c = nG$ ,  $G =$



Fig. 54.1. Cross-correlation of the above two selected Reed-Solomon codes shown in Fig.s 53 and 54. According to Mersereau & Siday (1981), there should be no more than 3 hits between any pair of time-frequency shifted patterns. The above codes were used only as *time-hopping* codes. Nonetheless, the maximum cross-correlation is 4, which is still excellent.

Reed-Solomon, n=[7,1,1]: p=23, Autocorr.

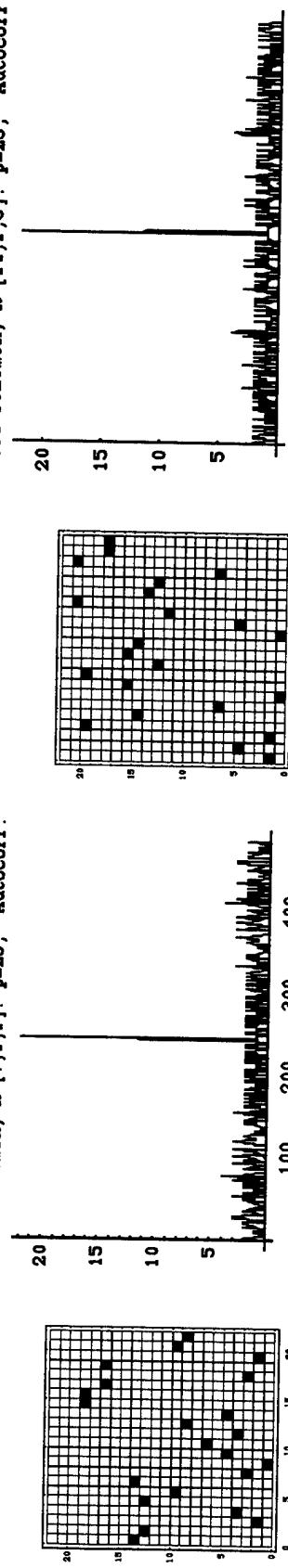


Fig. 55

Fig. 55. Reed-Solomon code, Example 1:  $c = nG$ ,  $G = \begin{bmatrix} I & I & I & \cdots \\ 5 & 5^2 & 5^3 & \cdots & I \\ 5^2 & 5^4 & 5^6 & \cdots & I \end{bmatrix}$ ,  $n = [7,1,1]$ ,  $p = 23$ .

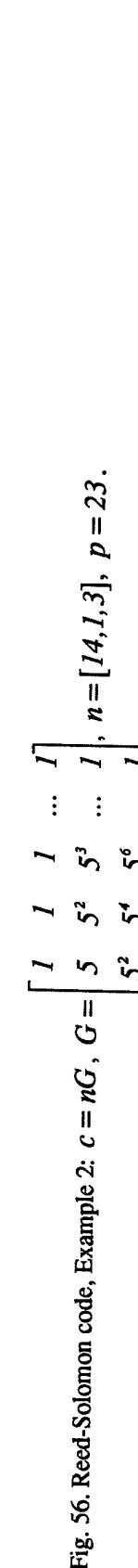


Fig. 56. Reed-Solomon code, Example 2:  $c = nG$ ,  $G = \begin{bmatrix} I & I & I & \cdots \\ 5 & 5^2 & 5^3 & \cdots & I \\ 5^2 & 5^4 & 5^6 & \cdots & I \end{bmatrix}$ ,  $n = [14,1,3]$ ,  $p = 23$ .

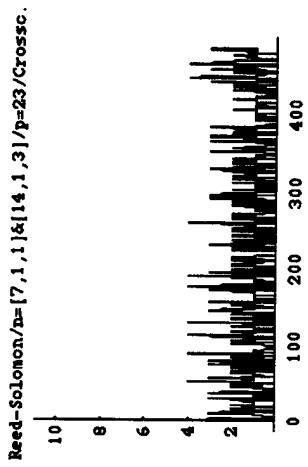


Fig. 56.1. Cross-correlation of the two selected Reed-Solomon codes shown in Fig.s 55 and 56. According to Mersereay & Seay (1981), there should be no more than 2 hits between any pair of *time-frequency* shifted patterns (for Doppler shifts of less than 4 chip bandwidths). The above codes were used only in *time-hopping* transmission. Nonetheless, the maximum cross-correlation is 4, which is still excellent.

Reed-Solomon #3, n=[1,2^10], p=64, Autocorr.

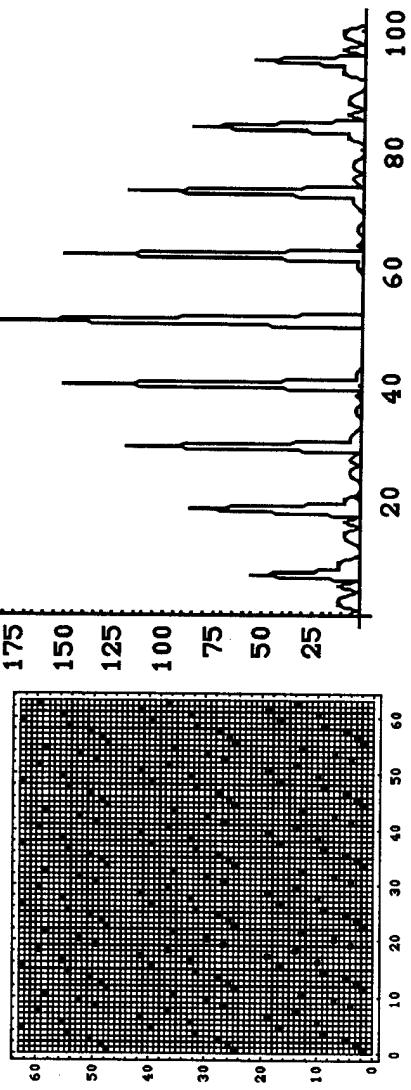


Fig.56.2. Reed-Solomon code:  $C = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 2 & 2^2 & 2^3 & \dots & 2 \end{bmatrix}$ ,  $n = [1, 2^7]$ ,  $p = 64$ . The auto-correlation performance is poor.

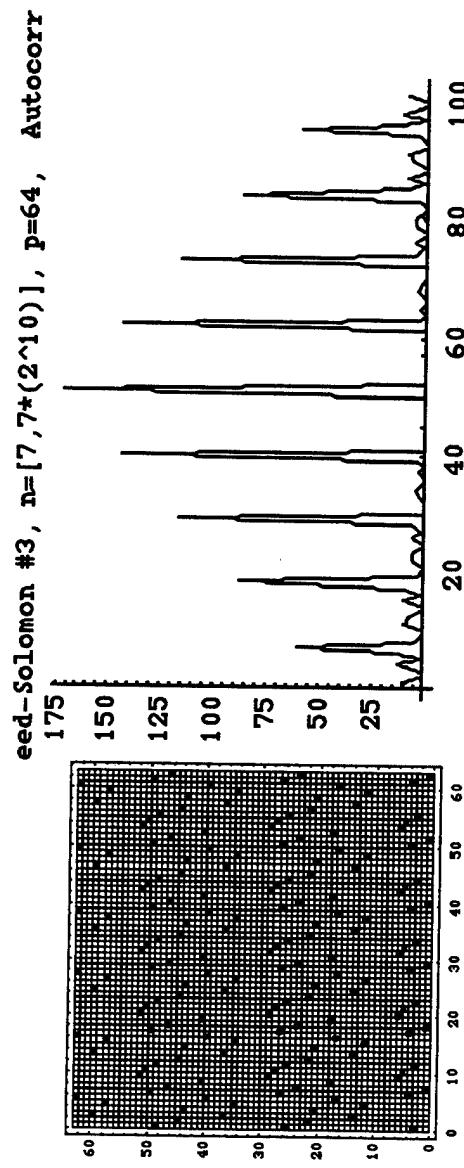


Fig. 56.3.  $c = nG$ ,  $G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 2 & 2^2 & 2^3 & \cdots & 2 \end{bmatrix}$ ,  $n = [7, 7 \times 2^{10}]$ ,  $p = 64$ . The auto-correlation performance is poor.

R-S #3,  $n = [1, 2^{10}] \& [7, 7*(2^{10})]$ ,  $p = 64$ , Crosscorr.

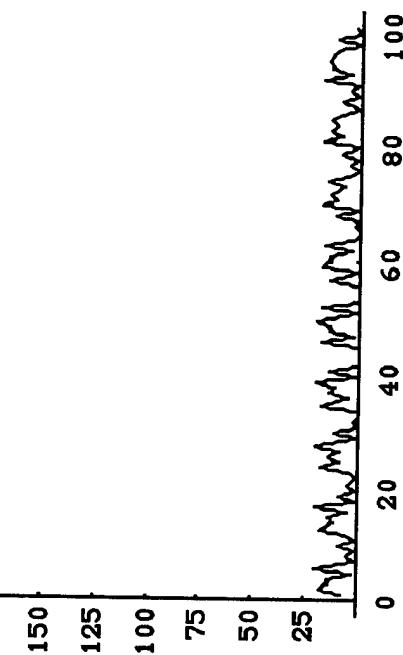


Fig. 56.4. Reed-Solomon, cross-correlation between the codes shown in Fig.s 56.2 and 56.3. The cross-correlation performance is excellent, but Fig.s 56.2 and 56.3 revealed poor auto-correlation performance for the two codes.

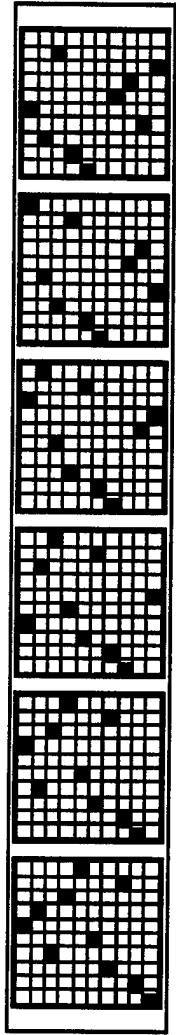


Fig. 56.5. Shaar & Davies<sup>30</sup>, Construction 1 (according to McEliece<sup>31</sup>).

Shaar & Davis #1, p=11, № 1, Autocorr.

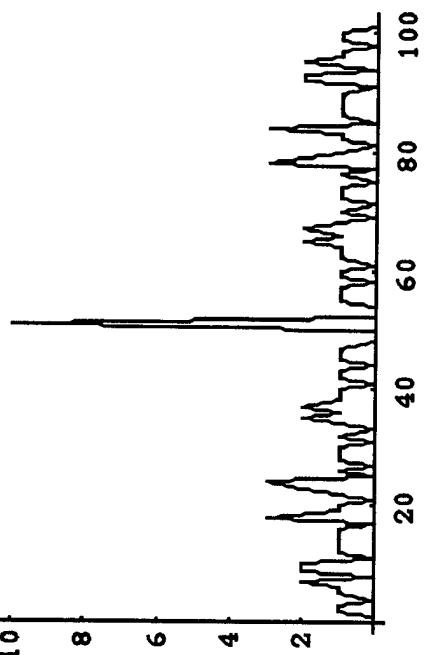


Fig. 56.6. Auto-correlation of Example 1 of Shaar & Davies, Construction 1 (according to McEliece). Maximum number of subband hits is 3.

<sup>30</sup>Shaar, A.A. & Davies, P.A., A survey of one-coincidence sequences for frequency-hopped spread-spectrum systems. *IEEE Proc.*, 131 Pt F, 719-724, 1984.

<sup>31</sup>McEliece, R.J., Some combinatorial aspects of spread spectrum communication systems. In J.K. Skwirzynski (Ed.), *New Concepts in Multi-User Communications*, NASI Stieltjes & Noordhoff, 1981.

Shaar &amp; Davis #1, p=11, No 2, Autocorr.

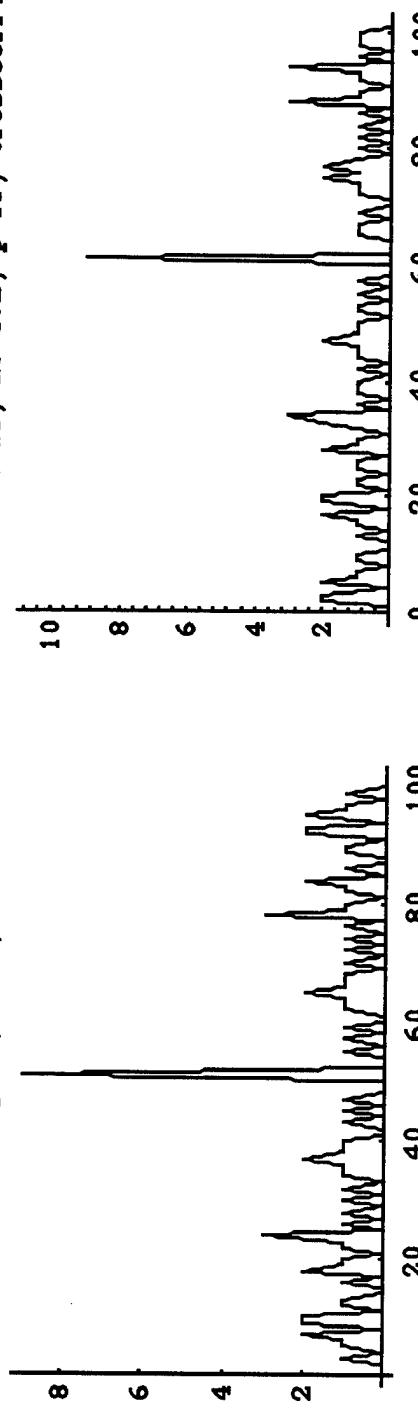


Fig. 56.7

Shaar &amp; Davis #1, No 1&amp;2, p=11, Crosscorr.

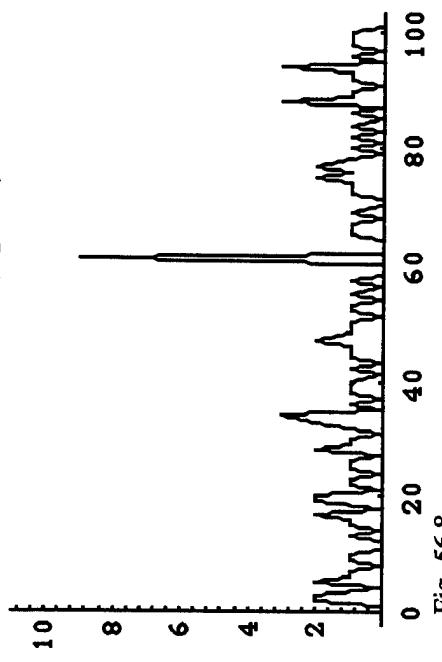


Fig. 56.8

Fig. 56.7. Auto-correlation of Example 2 of Shaar &amp; Davies, Construction 1 (according to McEliece). Maximum number of subband hits is 3.

Fig. 56.8. Cross-correlation of Examples 1 & 2 of Shaar & Davies, Construction 1 (according to McEliece). Maximum number of crossband hits is 9, indicating poor cross-correlation properties.

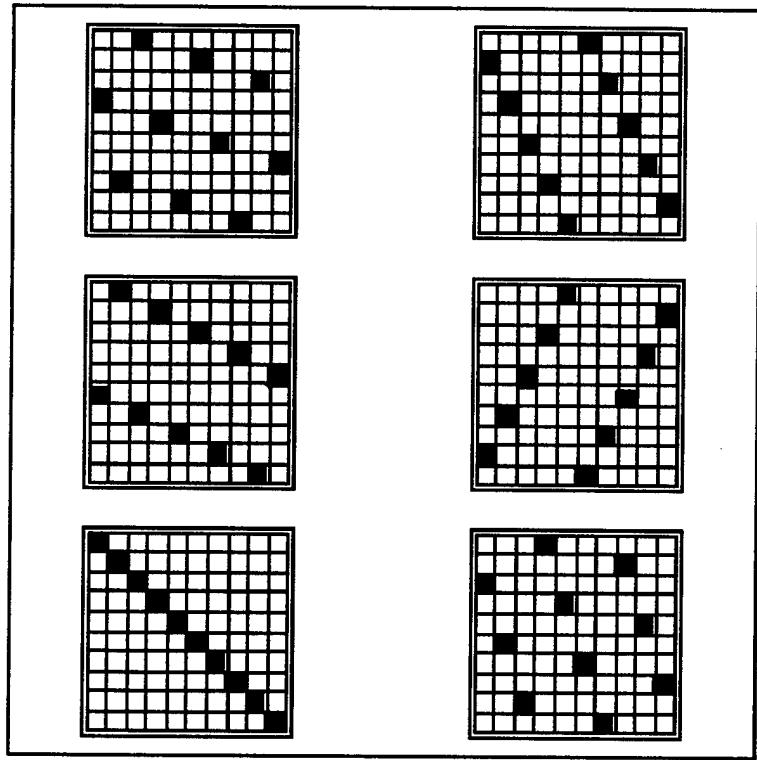


Fig. 56.9. Shaar & Davies<sup>32</sup>, Construction 2 (Linear congruences, and according to Shaar & Davies<sup>33</sup> and Titlebaum<sup>34</sup>).

<sup>32</sup>Shaar, A.A. & Davies, P.A., A survey of one-coincidence sequences for frequency-hopped spread-spectrum systems. *IEE Proc.*, 131 Pt F, 719-724, 1984.

<sup>33</sup>Shaar, A.A. & Davies, P.A., Prime sequences: quasi-optimal sequences for OR channel code division multiplexing. *Electron. Lett.*, 19, 888-890, 1983.

<sup>34</sup>Titlebaum, E.L., Time-frequency hop signals, Part I: Coding based upon the theory of linear congruences. *IEEE Trans. Aerospace & Electronic Systems*, AES-17, 490-493, 1981.

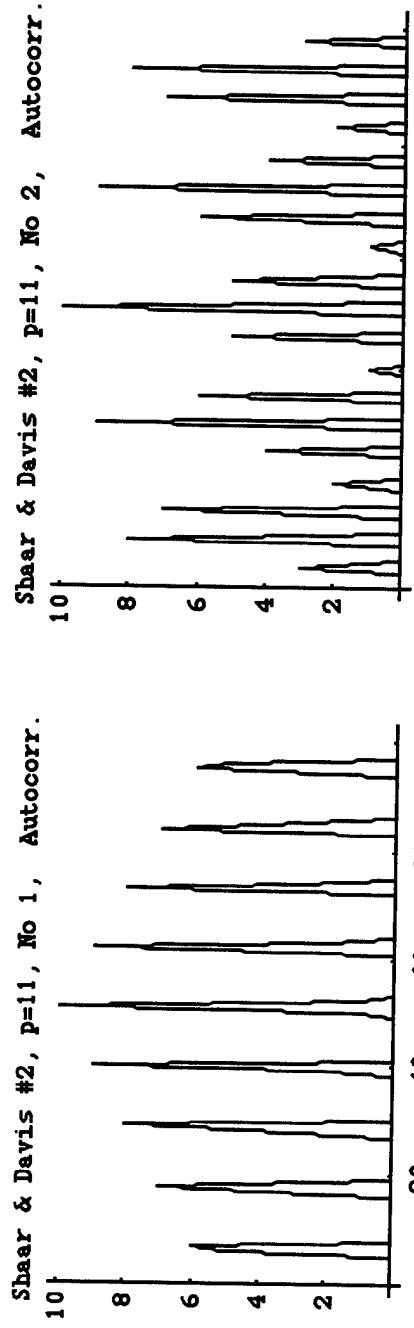


Fig. 56.10. Auto-correlation of Example 1 of Shaar & Davies, Construction 2 (Linear congruences and according to Shaar & Davies and Titlebaum). Poor auto-correlation is indicated.

Fig. 56.11. Auto-correlation of Example 2 of Shaar & Davies, Construction 2 (Linear congruences and according to Shaar & Davies and Titlebaum). Poor auto-correlation is again indicated.

Shaar&Davis #2, No 1&2, p=11, Crosscorr.

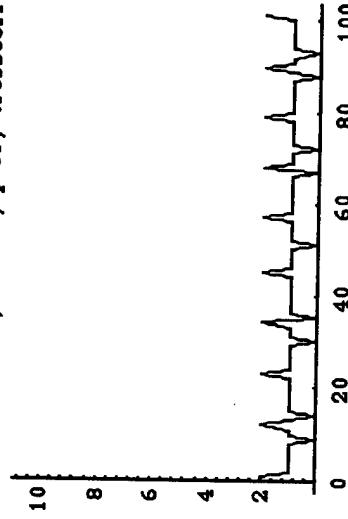


Fig. 56.12. Cross-correlation of Examples 1 & 2 of Shaar & Davies, Construction 2 (according to Shaar & Davies and Titlebaum). Maximum number of crossband hits is 2, indicating excellent cross-correlation properties.

We see that Construction 1 (Fig.s 56.5-56.8) and Construction 2 (Fig.s 56.9-56.12) of Shaar & Davies give opposite results - excellent auto-correlation properties and poor cross-correlation properties in the case of the former, and vice-versa in the case of the latter. In the next section, we analyze the reasons for this apparent tradeoff between auto- and cross-correlation properties. This leads us to consider the tradeoff between low out-of-phase auto-correlation and low cross-correlation:

The following analysis is based on Sarwate<sup>35</sup>, and adapted for use in the time hopping context (instead of the frequency hopping context). Let  $t$  denote the number of time slots in a frame of a TH system, and  $\tau_i$  denote the center frequency of the  $i$ 'th slot,  $1 \leq i \leq t$ , then a time hopping pattern is a sequence,

$$x = (x_1, x_2, \dots, x_T) \text{ or } x = (x_{\tau_1}, x_{\tau_2}, \dots, x_{\tau_T}) \quad (1)$$

of  $T$  elements from the set  $\{\tau_1, \tau_2, \dots, \tau_t\}$  of possible slot positions.  $x$  specifies the order in which the slots are to be used in a particular code. A composition vector,  $\mathbf{F}(x)$ , of the sequence  $x$  is defined as:

$$\mathbf{F}(x) = [F_1(x), F_2(x), \dots, F_t(x)], \quad (2)$$

where  $F_i(x)$  denotes the number of times that time positioning,  $\tau_i$ , is used or occupied in  $x$ . (For hyperbolic congruence codes and quadratic congruence codes, one time slot is occupied per frame, i.e.,  $F_i(x) = 1$ , for all  $i$ , and so for  $p=11$ ,  $\mathbf{F}(x) = \mathbf{10}$ ).

Therefore,

$$\sum_{i=1}^t F_i(x) = F. \quad (3)$$

(For hyperbolic congruence codes and quadratic congruence codes, one time slot is occupied per frame, i.e.,  $F_i(x) = 1$ , for all  $i$ , and so for  $p = 11$ ,  $F = 10$ ).

Furthermore<sup>36</sup>:

$$\sum_{i=0}^{t-1} F_i^2(x) = \|\mathbf{F}(x)\|^2 \geq F \left\lfloor \frac{F}{t} \right\rfloor + \left\lceil \frac{F}{t} \right\rceil (F \bmod t). \quad (4)^{37}$$

<sup>35</sup>Sarwate, D.V., Reed-Solomon Codes and the Design of Sequences for Spread-Spectrum Multiple-Access Communications. pp. 175-204, in Wicker, S.B. & Bhargava, V.K. (Eds) *Reed-Solomon Codes and Their Applications*, IEEE Press, 1994.

<sup>36</sup>Lempel, A. & Greenberger, H., Families of sequences with optimal Hamming correlation properties. *IEEE Trans. Information Theory*, IT-20, 90-94, 1974.

<sup>37</sup>The symbol  $\lfloor \cdot \rfloor$  is used for rounding downwards, i.e.,  $\lfloor x \rfloor = \max\{n \in \mathbb{Z} / n \leq x\}$ . Similarly, the symbol  $\lceil \cdot \rceil$  is used for rounding upwards.

(In the case of HCC and QCC, for  $p = 11$ ,  $F$  is an integer multiple of  $\frac{F}{t}$ , so no rounding downwards and upwards occurs, i.e.,  $F\left\lfloor \frac{F}{t} \right\rfloor + \left\lceil \frac{F}{t} \right\rceil = F\frac{F}{t}$ . Therefore:

$$\sum_{i=0}^{t-1} F_i^2(x) = \left\| \mathbf{F}(x) \right\|^2 \geq 10 \frac{10}{10} = 10. )$$

Equality holds in Eq. (4) if and only if:

$$F_i(x) = \left\lfloor \frac{F}{t} \right\rfloor \quad \text{for } (F \bmod t) \text{ values of } i, \quad (5)$$

$$F_i(x) = \left\lceil \frac{F}{t} \right\rceil \quad \text{for } t - (F \bmod t) \text{ values of } i. \quad (6)$$

(In the case of HCC and QCC, for  $p = 11$ ,  $F_i(x) = \left\lfloor \frac{10}{10} \right\rfloor = \left\lceil \frac{10}{10} \right\rceil = 1.$ )

The following apply under the specific magnitudes of  $F$  and  $t$ :

$$F\left\lfloor \frac{F}{t} \right\rfloor + \left\lceil \frac{F}{t} \right\rceil (F \bmod t) = F \quad \text{if } F < t, \quad (7)$$

also

$$F\left\lfloor \frac{F}{t} \right\rfloor + \left\lceil \frac{F}{t} \right\rceil (F \bmod t) \geq \frac{F^2}{t},$$

and

(8A & 8B)

$$F\left\lfloor \frac{F}{t} \right\rfloor + \left\lceil \frac{F}{t} \right\rceil (F \bmod t) = \frac{F^2}{t} \quad \text{iff } F \text{ is an integer multiple of } t.$$

(As, in the case of HCC and QCC  $F = t$ , then  $F\frac{F}{t} = \frac{F^2}{t} = \frac{10^2}{10} = 10 = F$ ).

Now the Hamming cross-correlation function<sup>38</sup> for two time hopping patterns,  $x$  and  $y$ , is:

$$H_{xy}(j) = \sum_{i=1}^F h(x_i, y_{i+j}), \quad 0 \leq j \leq F-1, \quad \text{and the sum } i+j \text{ is modulo } F, \quad (9)$$

and

---

<sup>38</sup>Lempel, A. & Greenberger, H., Families of sequences with optimal Hamming correlation properties. *IEEE Trans. Information Theory*, IT-20, 90-94, 1974.

$$h(a, b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{if } a \neq b. \end{cases} \quad (10)$$

Alternatively,

$$H_{x,y}(j) = F - d(x, S^j y), \text{ where} \quad (11)$$

$d(x, y)$  is the Hamming distance between sequences  $x$  and  $y$ ; and  $S$  is a one-place left shift operator. The Hamming autocorrelation function is  $H_{x,x}(j)$ .

If  $\bar{H}_{x,y}$  is the average Hamming cross-correlation value, and  $\bar{H}_{x,x}$  is the average out-of-phase Hamming autocorrelation, then:

$$F\bar{H}_{x,y} = \sum_{j=1}^F H_{x,y}(j) = \sum_{i=1}^t F_i(x)F_i(y) = \langle F(x), F(y) \rangle \quad (12)$$

and

$$(F-1)\bar{H}_{x,x} = \sum_{j=1}^F H_{x,x}(j) = \langle F(x), F(x) \rangle - F = \|F(x)\|^2 - F. \quad (13)$$

For any sequence:

$$\bar{H}_{x,x} > \frac{F}{F-1} \left( \frac{F}{t} - 1 \right), \quad (14)$$

(and for HCC and QCC,  $p=11$ , Eq. (14) indicates, trivially, that the average correlation  $\bar{H}_{x,x} > 0$ ).

The maximum out-of-phase correlation is:

$$\max_{0 < j < T} H_{x,x}(j) \geq \left\lceil \frac{F}{t} \right\rceil - 1. \quad (15)$$

Again, for HCC and QCC this is a trivial result, e.g., for  $p=11$ , Eq. (15) gives the maximum out-of-phase autocorrelation as greater than or equal to 9, when the in-phase autocorrelation is 10.

If  $\mathbf{X}$  is the set of  $k$  hopping patterns of period  $F$ , then:

$$\sum_{x \in \mathbf{X}} \sum_{y \in \mathbf{X}} \sum_{j=1}^F H_{x,y}(j) = \left\| \sum_{x \in \mathbf{X}} F(x) \right\|^2. \quad (16)$$

As:

$$\sum_{x \in \mathbf{X}} F_i(x) \geq 0 \text{ for all } i$$

and

$$\sum_{i=1}^t \sum_{x \in \mathbb{K}} F_i(x) = kF,$$

the right-hand is smallest when:

$$\sum_{x \in \mathbb{K}} F_i(x) = \left\lceil \frac{kF}{t} \right\rceil \text{ for } (kF \bmod t) \text{ values of } i, \quad (17)$$

$$\sum_{x \in \mathbb{K}} F_i(x) = \left\lfloor \frac{kF}{t} \right\rfloor \text{ for } t - (kF \bmod t) \text{ values of } i \quad (18)$$

(For HCC and QCC,  $p = 11$ , there are 10 orthogonal codes which can be generated, so  $k = 10$  and  $\frac{kF}{t} = 10$ .)

If

$\bar{H}_c(\mathbb{K})$  is the average value of  $H_{x,y}(j)$  over all  $k(k-1)$  hopping patterns  $x$  and  $y$  in  $\mathbb{K}$   
and all  $j, 1 \leq j < F$ ,

$H_c(\mathbb{K})$  is the maximum value,

$\bar{H}_a(\mathbb{K})$  is the average value of  $H_{x,x}(j)$  over all  $k(k-1)$  hopping patterns  $x \in \mathbb{K}$  and all  $j, 0 \leq j < F$

$H_a(\mathbb{K})$  is the maximum value,

$$\bar{H}_{\max}(\mathbb{K}) = \max\{\bar{H}_c(\mathbb{K}), \bar{H}_a(\mathbb{K})\},$$

$$H_{\max}(\mathbb{K}) = \max\{H_c(\mathbb{K}), H_a(\mathbb{K})\},$$

then from Eq. (16):

$$k(k-1)FH_c(\mathbb{K}) + k(F-1)H_a(\mathbb{K}) \geq k(k-1)F\bar{H}_c(\mathbb{K}) + k(F-1)\bar{H}_a(\mathbb{K}) \quad (19)$$

$$= \left\| \sum_{x \in \mathbb{K}} \mathbf{F}(x) \right\|^2 - kF \geq \frac{(kF)^2}{t} - kF.$$

(To put the result of Eq. (19) in perspective, for HCC and QCC,  $p = 11$ ,  
 $\frac{(kF)^2}{t} - kF = 900$ .)

If  $kF > t$ :

$$H_{\max}(\mathbb{K}) \geq \bar{H}_{\max}(\mathbb{K}) > \frac{F}{t} - \frac{1}{k}. \quad (20)$$

This is the lower bound of the maximum and average correlation all codes being used. (In the case of HCC and QCC,  $p = 11$ ,  $\frac{F}{t} - \frac{1}{k} = \frac{9}{10}$ .) However, this result is hardly useful.) A Singleton bound<sup>39</sup> is:

$$H_{\max}(\mathbf{x}) \geq \lceil \log_t(kF) \rceil - 1, \quad (21)$$

which, for HCC and QCC,  $p = 11$ , is  $H_{\max}(\mathbf{x}) \geq 1$ , and again, hardly useful.

What is useful, is to recast Eq. (19) in the form<sup>40</sup>:

$$H_c(\mathbf{x}) + \frac{1}{k-1} H_a(\mathbf{x}) \geq \bar{H}_c(\mathbf{x}) + \frac{1}{k-1} \bar{H}_a(\mathbf{x}) \geq \frac{1}{k-1} \left( \frac{kF}{t} - 1 \right), \quad (22)$$

making evident the tradeoff between the maximum or average crosscorrelation and the maximum or average autocorrelation. For example, for HCC and QCC,  $p = 11$ ,

$$H_c(\mathbf{x}) + \frac{1}{9} H_a(\mathbf{x}) \geq \bar{H}_c(\mathbf{x}) + \frac{1}{9} \bar{H}_a(\mathbf{x}) \geq 1.$$

The lower bound is not very helpful, but if the cross-correlation/auto-correlation - whether the maximum or the average - is decreased, then the auto-correlation/cross-correlation is increased. One should caution that, as different weights are attached to the cross- and the auto-correlation, the increases and decreases in either are not commensurate.

Adapting methods in Ref.<sup>41</sup> which addressed accuracy considerations in frequency hopping to time hopping, the time, accuracy and stability (TAS) of the transmitting and receiving system is equal to:

$$TAS_{(-3db)} \leq \frac{0.4}{\text{maximum slot position} \times \text{total number of frames}}, \text{ or}$$

$$TAS_{(-3db)} \leq \frac{0.4}{t \times T} = \frac{0.4}{t^2},$$

which, for  $p = 11$ , is:

$$TAS_{(-3db)} \leq \frac{0.4}{121} = 0.0033,$$

<sup>39</sup>van Lint, J.H., *A Course in Coding Theory*, Springer, Berlin, 1989;  
Sarwate, D.V., Reed-Solomon Codes and the Design of Sequences for Spread-Spectrum Multiple-Access Communications. pp. 175-204, in Wicker, S.B. & Bhargava, V.K. (Eds) *Reed-Solomon Codes and Their Applications*, IEEE Press, 1994.

<sup>40</sup>Sarwate, D.V., Reed-Solomon Codes and the Design of Sequences for Spread-Spectrum Multiple-Access Communications. pp. 175-204, in Wicker, S.B. & Bhargava, V.K. (Eds) *Reed-Solomon Codes and Their Applications*, IEEE Press, 1994.

<sup>41</sup>Albicki, A., Kinnen, E., Sobksi, A., Titlebaum, E. & Wronski, L.D., Distribution System Alarm Using a Multiple User Access Communication System: Transmitter and Receiver Design for Pilot Project, Phase I., Technical Report #1, June, 1992, Rochester Gas and Electric Corporation.

or, approximately three parts per 1,000. If a slot is defined as 1 psec., then this is 3 parts per 1,000 psec.s.

Defining, what is called in frequency hopping: a *hop expander*:

$$H = \frac{t T}{N^2},$$

then:

$$TAS_{(-3db)} \leq \frac{0.4}{HN^2}.$$

Extremely good correlation bounds can be obtained for both time hopping and frequency hopping (Fig. 58) - but not without a price. The timing accuracy plots for time hopping, equivalent to frequency accuracy plots for frequency hopping, are shown in Fig. 58. Clearly, for large  $N$  or large  $H$ , the timing accuracy required is high. Thus, increasing  $H$  results, mathematically, in better side-lobe-to-main-lobe ratios in auto-correlations and cross-correlations, but the price paid is in the hardware requirement of more stringent timing accuracy - as shown in Fig. 59. Indeed, Fig 60 shows that for a code of length 1,021, the  $TAS$  is  $10^{-14}$ , or 1 part per  $10^{14}$  slot durations.

However, the accuracy problem for time hopping is not as severe as that for frequency hopping. Whereas it is necessary for every frequency position to be recognized in frequency hopping, it is not necessary in the case of time hopping for the clock to set the timing of every slot. In fact the clock will not set every slot, but rather every frame, with the slots within the frame set by off-setting from the frame trigger. Therefore, the accuracy required in time hopping is set by the accuracy of the clock setting of the frame. Fig. 61 shows that in this case, for a code of length 1,021 the  $TAS$  is  $10^{-8}$ , or 1 part in  $10^8$  frame durations - an achievable accuracy.

## 1.5 Symmetry in Code Design

Fig.s 13-15 show, for the first time, symmetries underlying the orthogonal congruence codes and the Welch-Costas arrays. However, we have not yet been able to relate these revealed symmetries to the auto- and cross-correlation properties shown in Table 3. Fig.s 43-46 are based on spacing statistics and are more encouraging. The two codes which possess a superior mix of auto- and cross-correlation properties, the quartic and the quadratic congruence codes Fig.s 44 and 45, exhibit a more complex symmetry - a "symmetry modulation" - than do the cubic and hyperbolic congruence codes, Fig. 43, and the Welch Costas arrays, Fig. 46.

A surreal number<sup>42</sup> representation of these codes, Fig.s 47-51, is also preliminary, and the most that can be said is that the Welch-Costas arrays again reveal the simplest representation, perhaps correlated with the superior auto-correlation performance which is offset by the decidedly inferior cross-correlation performance, as shown in Table 3.

Turning to the Lexicographic codes, analyzed in Fig.s 39-41, the codes shown are all  $n = 7$ ,  $k = 4$  and of designed distance  $d = 3$ , but each lexicographic ordering is based on a different ordered basis, although each ordering provides the same code set. We see that the first (Fig. 39, Appendix Table 7) and the third (Fig. 41, Appendix Table 9) provide perfect left-right symmetry. The second (Fig. 40, Appendix Table 8) is not symmetrically ordered.

Fig.s 71-74 and 75-81 address changes in lexicographic ordering as the designed distance,  $d$ , is changed. Fig.s 71 and 76 (of the (7,4,3 ordering) stand out, exhibiting the greatest symmetry. However, as stated in the discussion of perfect codes, this code was not considered optimum, because the coderate,  $k/n$ , exceeds the channel capacity of 0.408. On the other hand, the code bounds derived are intended for *large n*. Therefore, further investigation might reveal that symmetry does play a role in choosing the optimum Lexicographic code.

---

<sup>42</sup> Knuth, D.E., *Surreal Numbers*, Addison-Wesley, Reading, MA, 1974; Conway, J.H., *On Numbers and Games*, Academic, New York, 1976.

## 2.0 Hardware Analysis

Presently, the type of optical switches being considered utilize optical interconnects with electronic control<sup>43</sup>. The optical switching fabric using such methods includes optical cross connects and optical multiplexers. Fully optical switching<sup>44</sup> remains in the laboratory, and requires optical random access memories, logic and control. Wavelength division multiplexing (WDM) is unique to photonics and was utilized recently in a demonstration<sup>45</sup>. The status of the photonics device technology has been recently reviewed<sup>46</sup>, but remains still thirty years behind that of electronics.

WDM is mostly accomplished with grating techniques. The use of active WDM, implementing opto-electronic conversion, for cross-connect applications can increase the capacity as well as the flexibility of networks. The first demonstration<sup>47</sup> of a managed multiwavelength network has been described.

All optical networks are ideal for implementation of network transparency. By transparency is meant that the user may choose any coding format and bit rate, i.e., independence from designated channels is obtained. However, all of this remains at the laboratory level, as presently optical networks are analog, noisy, nonlinear, nonregenerating and sensitive to crosstalk<sup>48</sup>, as opposed to electronic, digital, signal-restoring systems. It is presently believed that electronics and optics complement each other, so that an electronic bottleneck is encountered in transmission capacity, and a photonic bottleneck in channel granularity<sup>49</sup>.

Transmissions based on the synchronous digital hierarchy (SDH) and switching techniques such as asynchronous transfer mode (ATM) utilize electronic signal processing. However, it is felt that network solutions based on electronics will become increasingly complex and expensive. Therefore, the future lies with optics. Already the capacity of standard installed fiber is greater than is currently used - by a factor of 1000. This propels the search for optical device technology which will permit the exploitation of this capacity.

There are two major approaches to exploitation of the fiber capacity:

- wavelength division multiplexing (WDM) in which signals of different carrier frequencies are multiplexed onto a fiber; and

---

<sup>43</sup> Thylén, L., Karlsson, G. & Nilsson, Switching technologies for future guided wave optical networks: potentials and limitations of photonics and electronics. *IEEE Communications Magazine*, February, 1996, 106-113.

<sup>44</sup> Fork, R.L., *Phys. Rev. A*, 26, 2049, 1982; Smith, P.W., *Bell Syst. Tech. J.*, 1975-1993, 1982; Gibbs, H.M., *Optical Bistability: Controlling Light with Light*, Academic, New York, 1985.

<sup>45</sup> Dupraz, J., ATM: current status and perspectives of evolution. *Proc. ECOC '94*, Firenze, Italy, 1994, p. 555.

<sup>46</sup> Thylén, L., LiNbO<sub>3</sub> and semiconductor guided wave optics in switching and interconnects. In A. Marrakchi (ed), *Photonic Switching and Interconnects*, Marcel Dekker, New York, 1994.

<sup>47</sup> Hill, G. et al. A transport network layer based on optical network elements. *IEEE J. Lightwave Tech.*, 11, no 5/6, 667-79, 1993.

<sup>48</sup> Blumenthal, D. & Thylen, L., Coherent crosstalk induced BER floors in N x N space photonic switches. *Proc. IEEE/LEOS Topical Mtg on Optical Networks and Their Enabling Technologies*, Lake Tahoe, NV, 1994.

<sup>49</sup> Blumenthal, D., Channel capacity and bit-rate limitations of WDM multihop photonic switched all optical networks; Islam, M.N., *Ultrafast Fiber Switching Devices and Systems*, Cambridge University Press, 1992.

- ultra-high bit rate transmission (up to 100 Gb/s) using extremely short pulses generated by multiplexing optical tributary data streams or optical time division multiplexing (OTDM).

Both techniques can be used in switching and routing of signals. While all signal processing within telecommunication networks is currently performed electronically, WDM routing is well understood and can be implemented readily in the future with commercially available components. It is another story with OTDM, the components of which are still only laboratory demonstrated.

Presently, the upper limit of the number of WDM wavelengths is 10, but it is predicted<sup>50</sup> that numbers on the order of 100 will be required in the future for long distance (>3000 km) transmissions. For such requirements, fiber nonlinearities producing crosstalk will pose a constraint on the number of wavelengths used and over what distance they can be used. Two of the major nonlinear effects potentially interfering with transmission over long distances are four-wave mixing (FWM) and stimulated Raman scattering. Recently, compensation techniques have been used<sup>51</sup> to reduce FWM distortion to achieve 16 10-Gbs channels over 1000 km. However, currently there is no simple way to eliminate SRS.

The subcomponents necessary to implement OTDM is advancing. A mode-locked fiber-ring laser (ML-FRL) has been used to generate 3.5 ps pulses, which were then externally modulated and time multiplexed by 16 to generate a 100 Gbs pulse train<sup>52</sup>. This train was transmitted over 500 km of dispersion-shifted fiber. OTDM demultiplexing is necessary in which the demultiplexer is driven at the clock rate of the individual tributary channel and not at the full line rate. Either electro-optical, or all optical gating approaches are required for such demultiplexing. The fiber nonlinear optical loop mirror (NOLM) technique has been shown to produce subpicosecond multiplexing at 10 GHz repetition rate<sup>53</sup>.

It is envisaged that *a combination of WDM and OTDM networks* are required for transmission of *large numbers of wavelengths over large distances*. Such combinations would require the mapping of an *N*-channel WDM system into an *N*-channel OTDM system. Such multiplexers are beginning to be proposed (cf<sup>54</sup>).

Presently, there still exists a 3-order-of-magnitude mismatch between optical fiber and electronic device capacity. Whereas electronic devices and systems attain bit rates in the range of gigabits per second, photonic networks may exceeds 1 Tbits/sec. Therefore all optical methods are being developed to address this mismatch.

For example, the combination of spectral holography with conventional spatial Fourier transform holography permits the conversion of temporal signals into spatial signals and vice-versa<sup>55</sup>. These conversions can be realized in real time, providing the possibility of all-optical time division multiplexing and demultiplexing. Time-to-space

<sup>50</sup> O'Mahony, M.J., Optical multiplexing in fiber networks: progress in WDM and OTDM. *IEEE Communications Magazine*, December, 1995, pp. 82-88.

<sup>51</sup> Oda, K., *Proc. Opt. Fiber Conf. (OFC '95)*, San Diego, Feb., 1995, PD22.2-22.5.

<sup>52</sup> H., Kawanishi, S. & Uchiyama, K., *Proc. Opt. Fiber Conf. (OFC '94)*, paper no TuD5.

<sup>53</sup> Yamada, E., Suzuki, K. & Nakazawa, M., *Proc. Opt. Fiber Conf. (OFC '95)* San Diego, pp. 289-90.

<sup>54</sup> Lacey, J.P.R. et al., All-optical WDM to TDM transmultiplexer. *Electron. Lett.*, 30, 1612-13, 1994.

<sup>55</sup> Masurenko, Y.T., *Opt. Spectrosc.*, 57, 1, 1984;

Ema, K., Kuwata-Gonokami, M. & Shimizu, F., *Appl. Phys. Lett.*, 59, 2799, 1990;

Nuss, M.C., Li, M., Chiu, T.H., Weiner, A.M. & Partovi, A., *Opt. Lett.*, 19, 664, 1994.

conversions of ultrashort light pulses have been demonstrated with four-wave<sup>56</sup> and three-wave<sup>57</sup> interactions. Holographic optical processing which permits parallel-to-serial (space-to-time) optical signal conversion has also been demonstrated<sup>58</sup>.

Sun et al<sup>59</sup> have demonstrated a long distance interferometer (LDI) providing interferometric interaction between transmitter and receiver nodes in remote locations. An LDI permits phase information coded beam and the reference beam from a transmitter node to be transmitted to a remote receiver node. The long distance channel must preserve coherence in the beams for accurate multiplexing and demultiplexing to occur. The Sun et al method uses acousto-optic devices to achieve frequency division of two optical waves. The interferometer consists of two Mach-Zehnder-type devices, one at the transmitter, and the other at the receiver node, connected by a single optical communication channel. The Mach-Zehnder devices are implemented by acousto-optic Bragg cells, a phase modulator and a beam splitter. The disadvantage of the system is that it requires high-precision phase locking of two RF signals to achieve identical frequency shifts at both acousto-optical modulators.

The conclusion is that the field of high data rate communications is moving toward a combination of WDM and OTDM networks. The devices necessary

- to service OTDM,
- to interface WDM and OTDM,
- to interface optics and electronics,

are still under development in the laboratory, but a proof-of-concept demonstration is feasible. Of greater importance to the present endeavor: both WDM and OTDM will require optical codes of the type addressed in the present work. Weiner (1995)<sup>60</sup> is extremely useful in providing a broad background to these trends. Quite apart from the capability of generating optical pulses on the order of 10 femtoseconds (6 femtoseconds is the world record), there is also the revolutionary capability of ultrafast optical waveform generation. This bears directly on what codes can be technically implemented, in that both wavelet transmissions and several bit word "macropulses" (ultrafast waveform synthesis) can be achieved.

The methods available for the synthesis of optical waveforms include the "zero-dispersion pulse compressor" which achieves the said waveform synthesis by parallel modulation of the optical spectral components that make up the ultrashort pulse<sup>61</sup>. Another method is the "fiber and grating pulse compressor" in which the chirp of a pulse exiting a fiber is compensated by the temporal dispersion of gratings<sup>62</sup>. A very simple, but effective method for achieving pulse shaping employs an amplitude mask consisting of two slits. Using two isolated spectral components an interference in time occurs producing a high

<sup>56</sup>Ema, K., Kuwata-Gonokami, M. & Shimizu, F., *Appl. Phys. Lett.*, 59, 2799, 1990;  
Nuss, M.C., Li, M., Chiu, T.H., Weiner, A.M. & Partovi, A., *Opt. Lett.*, 19, 664, 1994.

<sup>57</sup>Mazurenko, Y.T., Spiro, A.S., Putilin, S.E. and Beliaev, A.G., *Opt. Spectrosc.*, 78, 122, 1995.

<sup>58</sup>Sun, P.C., Mazurenko, Y.T., Chang, W.S.C., Yu, P.K.L. & Fainman, Y., All-optical parallel-to-serial conversion by holographic spatial-to-temporal frequency encoding. *Optics Letters*, 20, 1728-1730, 1995.

<sup>59</sup>Sun, P.C., Mazurenko, Y. & Fainman, Y., Long-distance frequency-division interferometer for communication and quantum cryptography. *Optics Letters*, 20, 1062-1064, 1995.

<sup>60</sup>Weiner, A.M., Femtosecond optical pulse shaping and processing. *Progress in Quantum Electronics* 19, 161-237, 1995.

<sup>61</sup>Froehly, C., Colombeau, B. & Vampouille, M., in *Progress in Optics*, E. Wolf (ed), North-Holland, Amsterdam, vol. 20, p. 65, 1983;

Martinez, O.E., *IEEE J. Quantum Electronics* 23, 59, 1987.

<sup>62</sup>Grischkowsky, D. & Balant, A.C., *Appl. Phys. Lett.*, 41, 1, 1982.

frequency tone burst<sup>63</sup> (or a several bit word macropulse). The phase of the temporal beat note can also be precisely controlled by a phase mask. This method is the temporal analog of Young's two-slit interference experiment.

The pulse shaping masks achieving these results include:

- Fixed phase and amplitude masks fabricated using microlithographic techniques.
- Spatial light modulators:
  - liquid crystal
  - acousto-optic deflectors
  - modulator arrays based on III-V semiconductor devices
- Movable and deformable mirrors.
- Holographic masks.
- Amplifying media.
- Passive pulse shaping techniques
  - delay lines and interferometers
  - volume holograms
  - acousto-optic filters
  - resonant molecules
- SEED arrays<sup>64</sup>
- Semiconductor electro-optic modulators based on the quantum confined Stark effect (future).

Multi-element phase modulators used for gray-level phase control can also be used for pulse position modulation, as phasing sweeps correspond in the time domain to positioning of pulses. This is because if

$$f(t) \text{ and } F(\omega)$$

are a Fourier transform pair, then the delayed signal and its Fourier transform:

$$f(t - \tau) \text{ and } F(\omega) \exp(-i\omega\tau)$$

are also pairs, revealing that a pulse can be retarded, or advanced, by imposing a linear phase sweep onto its spectrum.

Pulse position modulation can be achieved over many pulse widths by using this kind of spectral phase manipulation<sup>65</sup>. The phase change per (temporal) pixel is 0 and  $\pm\pi/4$  and the output pulses occur at 0 and  $\pm 1.38$  psec. The only limitation is that the phase change must be less than  $\pi$ . Grey-level phase control can also be achieved by programmable compression of chirped optical pulses. By using a multielement phase element within a pulse shaping apparatus, it is possible to adjust the magnitude and the sign of a phase sweep electronically without moving parts<sup>66</sup>.

Liquid crystal modulator arrays have also been used with chirped pulse amplifiers to achieve programmably shaped high power pulses at the millijoule level<sup>67</sup>.

<sup>63</sup>Weiner, A.M., Heritage, J.P. & Kirschner, E.M., *J. Opt. Soc. Am.*, B5, 364, 1988.

<sup>64</sup>Nuss, M.C., Knox, W.H. & Miller, D.A.B., *Electron Lett.* 1995.

<sup>65</sup>Weiner, A.M., Leaird, D.E., Patel, J.S. & Wullert, J.R., *IEEE J. Quantum Electronics* 28, 908, 1992.

<sup>66</sup>Treacy, A.E., *IEEE J. Quantum Electronics* 5, 454, 1969.

<sup>67</sup>Pinkos, D., Squier, J., Kohler, B., Yakovlev, V.V., Wilson, K.R., Schumaker, D. & Bucksbaum, P., Production of programmable amplified, shaped pulses in femtosecond lasers. *Ultrafast Phenomena IX*, Dana Point, CA, 1994.

Binary phase masks can be fabricated to generate repetitions of code sequences. For example, Weiner & Leaird have generated  $m$ , or maximal length, sequences<sup>68</sup>. Furthermore, pulse trains with repetition rates as high as 12.5 THz have been generated by phase-only filtering using 22 fsec input pulses<sup>69</sup>.

Another method of promise which can be used to generate  $m$ -ary words per slot in communications is a "time domain lens" which employs electro-optic phase modulators. The time domain lens is based on the analogy between dispersion in the time domain and diffraction in the spatial domain, and also between quadratic phase modulation in time and the action of a thin lens in space<sup>70</sup>. Such temporal imaging systems can provide:

- expansion and compression in the time domain
- waveform synthesis
- time-reversal
- waveform measurement

Holographic methods have been used for pattern recognition<sup>71</sup>. These methods are now being used for:

- recall of shaped waveforms
- time-reversal
- matched filtering
- correlation
- convolution
- dispersion compensation (as in fiber optic communications).

A possible use for these holographic methods is wavelet pulse shaping and the generation of  $m$ -ary pulse forms. Second, or higher-order diffraction in spectral holography makes possible new signal processing applications involving multiple correlations<sup>72</sup>. Furthermore, second-order diffraction can be used to implement a simplified associative memory for spatial images<sup>73</sup>. As pointed out by Weiner<sup>74</sup>, the possibility is then raised of constructing holograms for simple associative memories for ultrafast pulse sequences, in which a complete output sequence might be generated in response to a partial input pulse sequence. Even more, the partial input pulse could generate an  $m$ -ary pulse within the pulse sequence.

Mazurenko has proposed time domain multiplexing using spectral holography and hybrid space-time conversions<sup>75</sup>. In this scheme a one dimensional array of optical fibers carries separate data channels, which are synchronized and operate at identical bit rates. Once every data cycle a Fourier transform of the image created by the (spatially) parallel data stream is written using a reference pulse synchronized to the data channels. This hologram is read out once per cycle by an ultrashort test pulse. Therefore a time-division multiplexing occurs, in which the spatially parallel input channels are converted a high

<sup>68</sup>Weiner, A.M. & Leaird, D.E., *Optics Letters* 15, 51, 1990.

<sup>69</sup>Reitze, D.H., Weiner, A.M. & Leaird, D.E., *Applied Physics Letters* 61, 1260, 1992.

<sup>70</sup>Kolner, B.H. & Nazarathy, M., *Opt. Lett.*, 14, 630, 1989.

<sup>71</sup>Collier, R., Burckhardt, C.B. & Lin, L.H., *Optical Holography*, Academic Press, 1971.

<sup>72</sup>Weiner, A.M. & Leaird, D.E., *Optics Lett.*, 19, 123, 1994.

<sup>73</sup>Paek, E.G. & Jung, E.C., *Optics Lett.*, 16, 1034, 1991.

<sup>74</sup>Weiner, A.M., Femtosecond optical pulse shaping and processing. *Progress in Quantum Electronics* 19, p. 64, 1995.

<sup>75</sup>Mazurenko, Y.T., *Opt. Eng.*, 31, 739, 1992.

speed serial bit sequence. The inverse process of time-to-space conversion from a spectral hologram with a readout using a monochromatic continuous-wave laser, is also possible.

Photon echo experiments on the picosecond time scale have been performed in which both spatial and temporal holographic processing are performed simultaneously<sup>76</sup>. This could permit the recording of four-dimensional holograms for associative memory applications.

The present maturity of device technology permits a few conclusions:

- Although point-to-point optical transmissions using TDM of 100 Gbit/sec have been demonstrated, the capability of maintaining the required precise synchronism in a geographically distributed network has yet to be developed. Pulse shaping could be used for address and data encryption so that interleaved optical packet transmission might be an alternative to timing methods.
- WDM operates with separate wavelength channels, each operating at bit rates up to Gbits/sec and aggregate rates far higher. The current need of WDM networks is for a multiplicity of laser sources providing the different wavelengths for the WDM channels. Furthermore, the wavelengths need to be stabilized especially when the laser sources are independent and geographically separated.
- In contrast, CDMA can support tens to hundreds of simultaneous users for aggregate throughputs of hundreds of Gbits/sec<sup>77</sup>. *A key advantage is that device operating speeds and synchronization are required only at the individual bit rates, not at the aggregate rate.* Furthermore, CDMA can provide a much larger address space than WDM, as a larger number of CDMA codes are available than WDM wavelengths. A challenge for implementing CDMA is the need for dispersion compensation.

The encoding and decoding of CDMA into phase patterns has been demonstrated<sup>78</sup> using phase masks. The same technique has been used for larger contrasts in maximum length sequences<sup>79</sup>. Weiner has also pointed out that pulse shaping can be used to achieve the dispersion compensation necessary for CDMA over long distances<sup>80</sup>. A constant (wavelength independent) dispersion can be compensated by adjusting the grating separation in pulse encoding/decoding. Programmable phase modulators or deformable mirrors can also be used as pulse shaping masks in order to compensate for dispersion. Furthermore, holographic methods can implement self-aligning compensation capabilities, in which the system automatically adjusts to match the dispersion of the incoming signal.

Zaccarin, D. & Kavehrad, M. have proposed a CDMA system in which encoding is achieved by means of spectral amplitude filtering with a pulse shaping apparatus<sup>81</sup>. Only a broadband light source is required which means that fiber dispersion is less important. Encoding and decoding is performed by amplitude masks. It is estimated that for a maximal length code of length 511, the system would support 200 users at 500 Mbit/sec at a  $10^{-9}$  error rate.

<sup>76</sup>Saari, P., Kaarli, R. & Ratsep, M., *J. Luminescence* 56, 175, 1993.

<sup>77</sup>Salehi, J.A., Weiner, A.M. & Heritage, J.P., *J. Lightwave Tech.* 8, 478, 1990.

<sup>78</sup>Weiner, A.M., Heritage, J.P. & Salehi, J.A., *Opt. Lett.*, 13, 300, 1988.

<sup>79</sup>Weiner, A.M., Heritage, J.P. & Kirschner, E.M., *J. Opt. Soc. Am.*, B5, 1563, 1988.

<sup>80</sup>Weiner, A.M., Femtosecond optical pulse shaping and processing. *Progress in Quantum Electronics* 19, p. 87, 1995.

<sup>81</sup>Zaccarin, D. & Kavehrad, M. *IEEE Phot. Tech. Lett.*, 5, 479-482, 1993.

Other approaches to CDMA include the use of various types of fiber optic tapped delay lines<sup>82</sup>.

### 3. Recommendations

The major findings and recommendations of this study are:

- The work over the past 12 months has shown the benefit of a *two-stage hierarchy* of codes. At the first stage, *orthogonal codes* define the microchannel or user, identified by a first (temporal) matched filter. Once a code is identified by this filter, a second (temporal) matched filter identifies the pulse-position-modulated data with error correction, hence *error-correcting codes* are required for the second stage. The recommendations of this study are that congruence codes be used as orthogonal codes (hyperbolic, quadratic, cubic and quartic) and Lexicographic codes for error correction codes.
- Besides a *hierarchical backbone topology*, a *multidimensional coding scheme* intermixing CDMA, TDMA and WDM provides the possibility of the highest data rates.
- These recommended techniques provide signal spreading techniques, but they are *spread time techniques*, as opposed to spread spectrum techniques.
- A major part of this study addressed the impact of *symmetry principles* on code design to achieve optimum properties. This study provides many examples of heretofore unknown symmetries underlying code design. Future work will address the use of symmetry in designing optimum codes.

The unique approaches taken in this study were:

- (1) *spread time techniques* (as opposed to spread spectrum techniques), which permit the highest data rates with the highest S/N and exploit the availability of optical pulse technology and the recent capabilities in pulse crafting and holography;
- (2) *Hierarchical* backbone communications link topologies, with *multidimensional* coding schemes;
- (3) the incorporation of WDM as well as wavelet diversity with respect to scale and translation when possible;
- (4) the use of *symmetry principles* for insight into optimum coding principles;
- (5) *Lexicographic ordering* for perfect code generation.

The results of a successful implementation of these recommendations will be:

- maximum permissible data rate transmissions will have been achieved using the newest device technologies;
- multi-dimensional coding based on hopping principles will have been demonstrated;
- the techniques of CDMA and TDMA will have been linked for optimum system usage ;

---

<sup>82</sup>For example: Griffin, R.A., Sampson, D.D. & Jackson, D.A., *Photonics Technology Letters* 4, 1401, 1992.

- spread time principles, as opposed to spread spectrum principles, will have been demonstrated for the first time as the method of choice for the highest data rate communications.

#### 4. Future Work

The technical problems addressed in future work are:

- In the case of optical orthogonal codes: the optimum tradeoff needs to be achieved between cross- and auto-correlation properties.
- In the case of error correcting codes: algorithmic methods need refining for code generation;
- The relationship needs defining between orthogonal and error correcting codes.

The approaches taken differ from all other approaches in that:

- optimum code designs are sought based on symmetry principles;
- unifying principles are sought between orthogonal and error correcting codes;
- hierarchical, spread time systems are proposed utilizing both orthogonal codes for channel definition and error correcting codes for data encryption;
- algorithmic lexicographic ordering is sought for perfect code generation.

A fruitful area of future research lies in investigating codes generated by lexicographic ordering. For example, all three of the Lexicographic-Greedy (7,4) codes of Appendix Tables 7-9 have the same generator matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

because all three code generations, based on different basis vectors, produce the same code words (i.e., the code vectors for which  $g = 0$ ). However, the plots, Tables 39-42 above, of the lexicographic ordering of these codes according to the different  $g$ -values, i.e., the ordering of the coset vectors or noncode words, indicate (1) symmetries underlying that ordering, and (2) differences between the three methods of code generation due to the differences in choice in basis vectors. The standard arrays for the three codes, Appendix Tables 4-6, also reveal that whereas in the case of the (7,4) code shown in Appendix Table 7 the coset leaders are also the basis vectors, this is not so for the codes shown in Appendix Tables 8 and 9, even although the codes themselves generated from all three sets of basis vectors are the same. This finding will have a bearing on the design of efficient error correction codes and will be investigated further.

Future work will also address:

- Frequency and Wavelet diversity: The present results are generic and need interpretation within the context of actual device capability. Presently only 4 wavelengths are available, and the diversity of wavelets in terms of scale and orthogonality has not been determined.
- Symmetries underlying optimum code design: These discoveries need find use in code design, i.e., investigations based on symmetry principles.
- Cross- and auto-correlation bounds: Bounds need to be tied to code design in a predictive way.
- Probability of error analysis indicates that error decreases with (1) an increase in the matched filter threshold, and (2) an increase in code length: Receiver threshold properties need to be analyzed together with code properties.
- $\lambda = 2$  optical orthogonal codes may be as good as  $\lambda = 1$  codes with receiver hard limiting: The tradeoffs between the properties of optical orthogonal codes with hard-limiting and with hard-limiting, need to be investigated.
- Lexicographic-Greedy codes: The discovered  $g$ -value symmetries can be used in code design.
- Lexicographic-Greedy codes: The Greedy code algorithm needs development for longer code generation.
- Code rate  $k/n$  bounds: The code bound derivations are for large  $n$ . The bound needs to be derived for all  $n$  and the result related to symmetry principles in optimum code design.

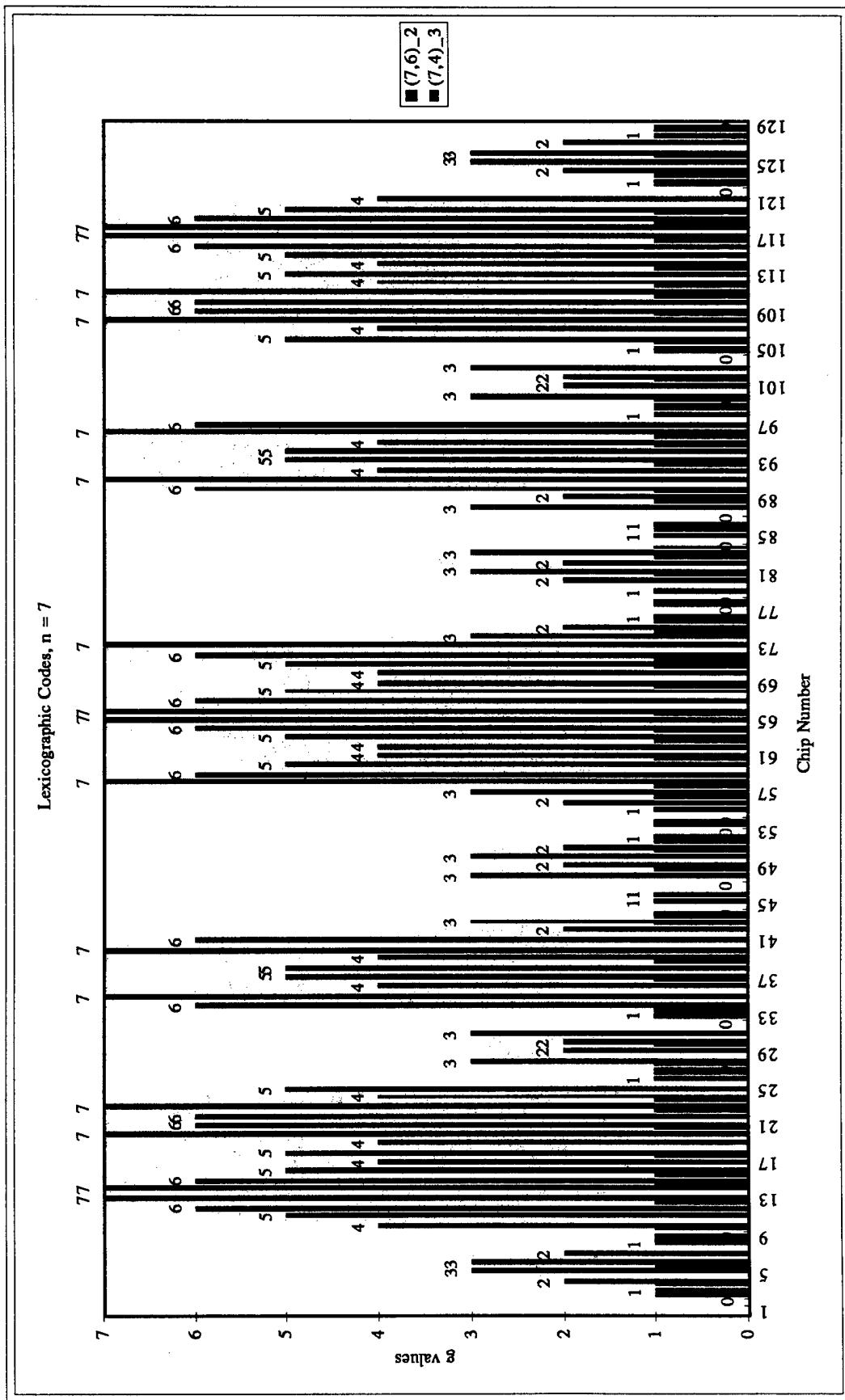


Fig. 71.  $g$  value representation of Lexicographic Codes,  $(n, k, d) = (7, 6, 2)$  (Appendix: Table 10) and  $(7, 4, 3)$  (Appendix: Table 11).

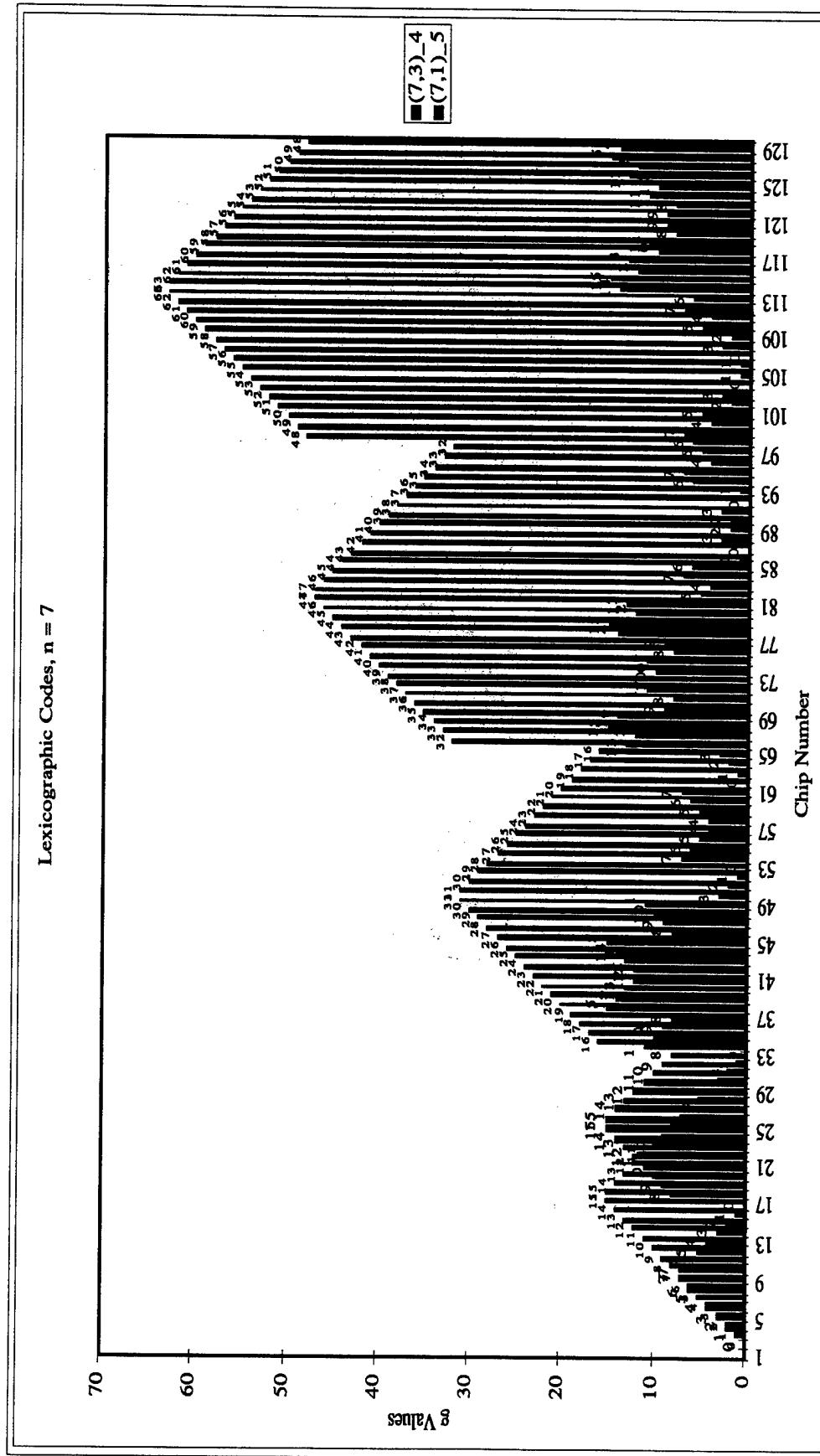


Fig. 72.  $g$  value representation of Lexicographic Codes,  $(n,k,d) = (7,3,4)$  (Appendix: Table 12) and  $(7,1,5)$  (Appendix: Table 13).

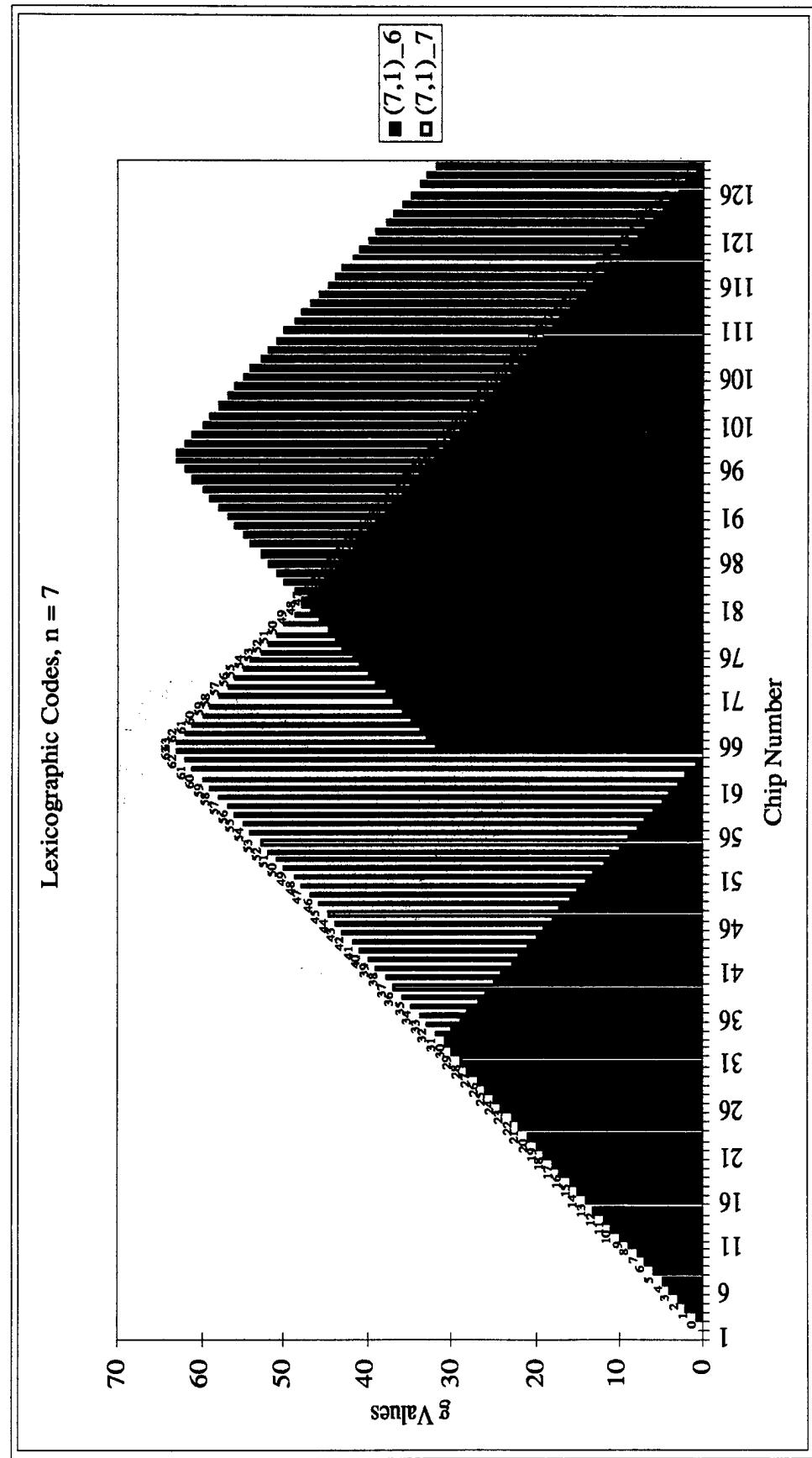


Fig. 73.  $g$  value representation of Lexicographic Codes,  $(n,k,d) = (7,1,6)$  (Appendix: Table 14) and  $(7,1,7)$  (Appendix: Table 15).

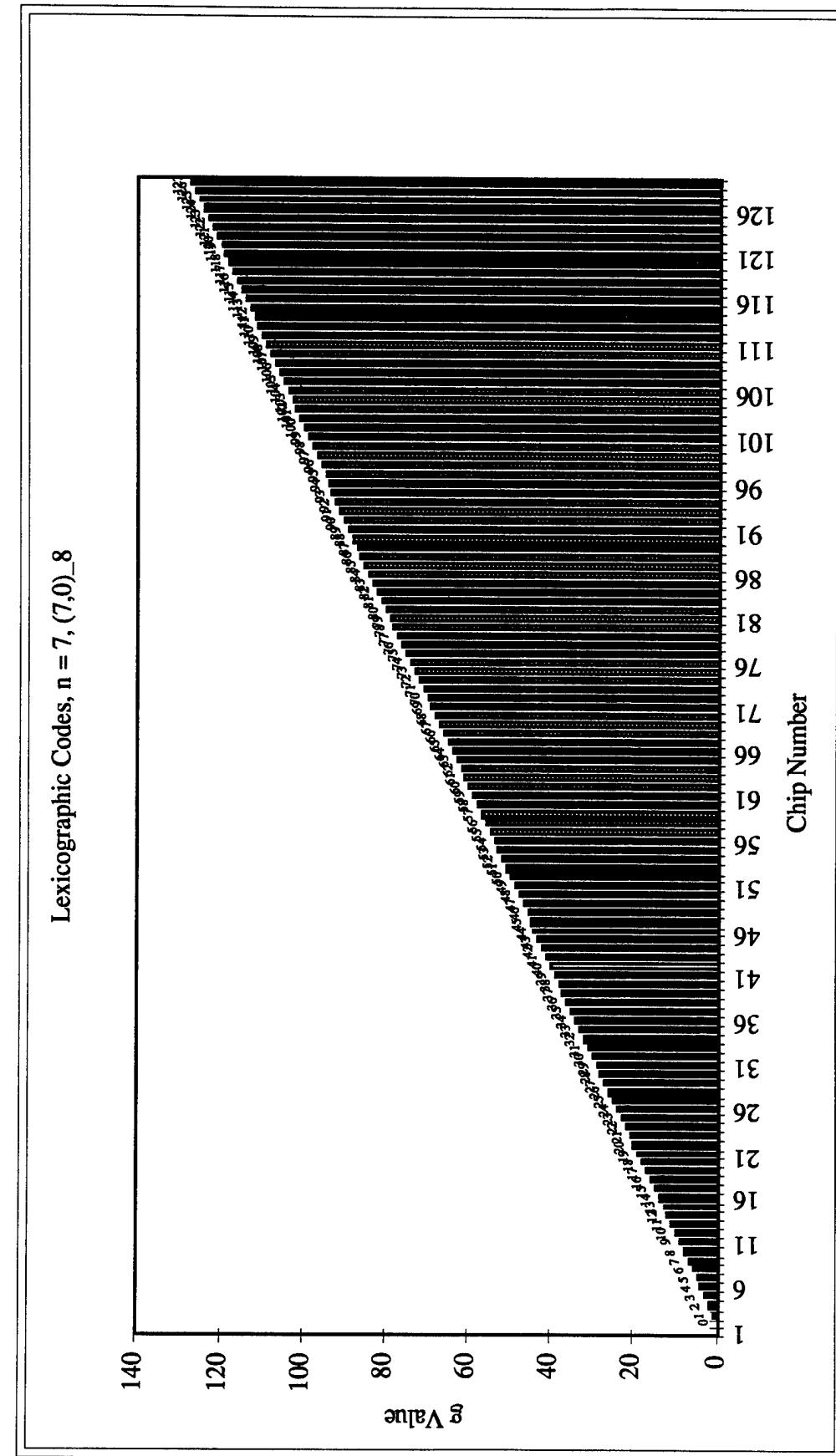


Fig. 74.  $g$  value representation of Lexicographic Codes,  $(n,k,d) = (7,0,8)$  (Appendix: Table 16).

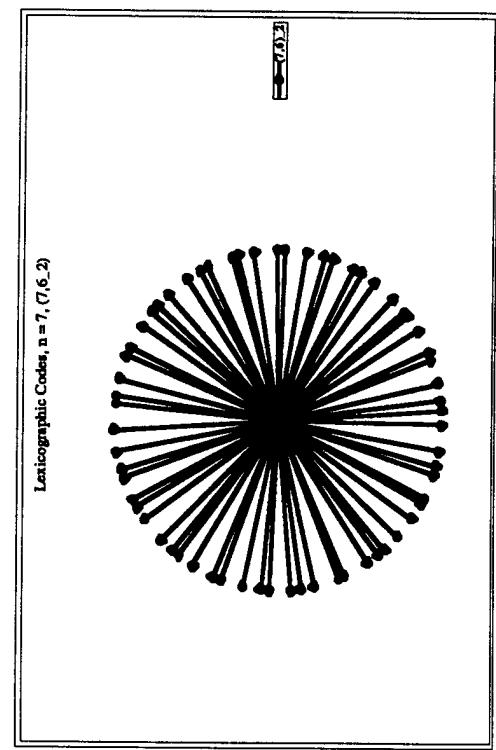


Fig. 75. Lexicographic Code, 7,6,2 (Table 10). All  $g$  values are represented radially.

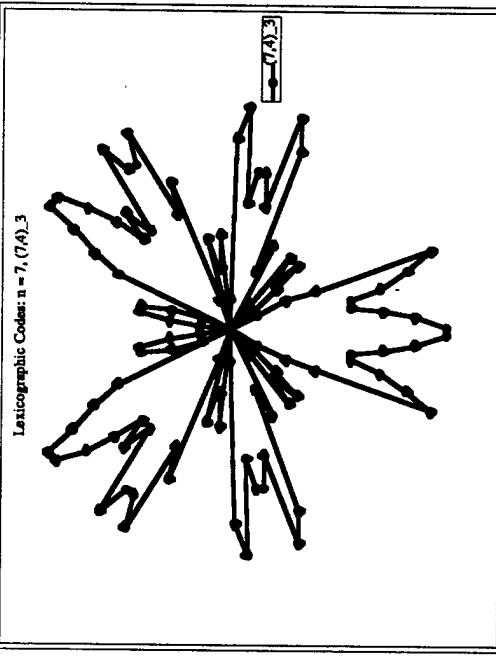


Fig. 76. Lexicographic Code, 7,4,3 (Table 11).

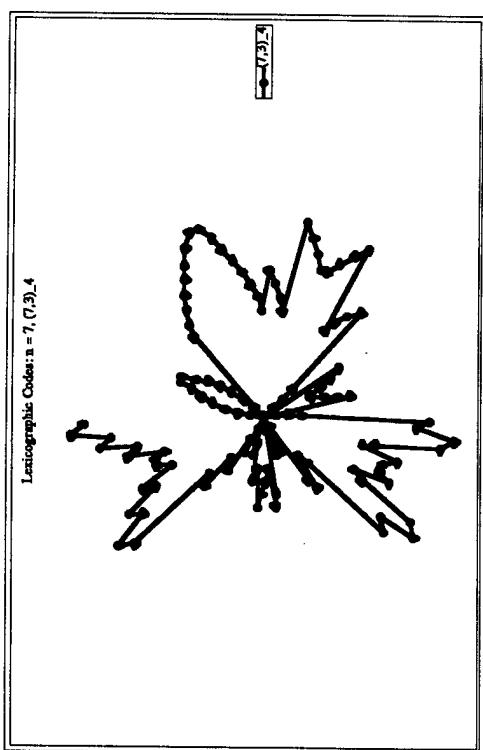


Fig. 77. Lexicographic Code, 7,3,4 (Table 12).

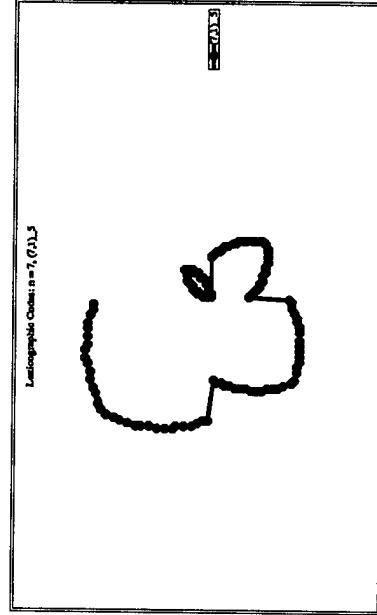


Fig. 78. Lexicographic Code, 7,1,5 (Table 13).

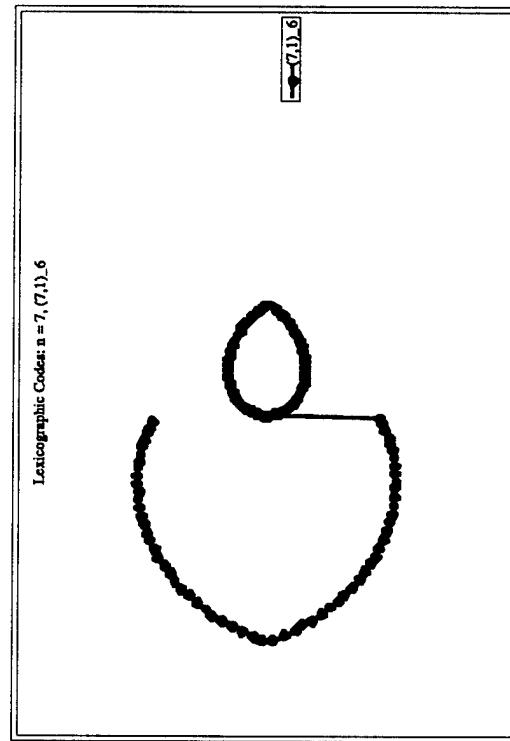


Fig. 79. Lexicographic Code, 7,1,6 (Table 14). All  $g$  values are represented radially. Fig. 80 Lexicographic Code, 7,1,7 (Table 15).

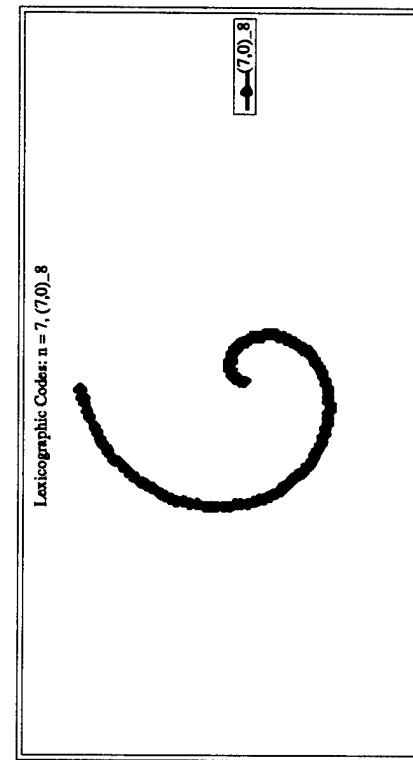
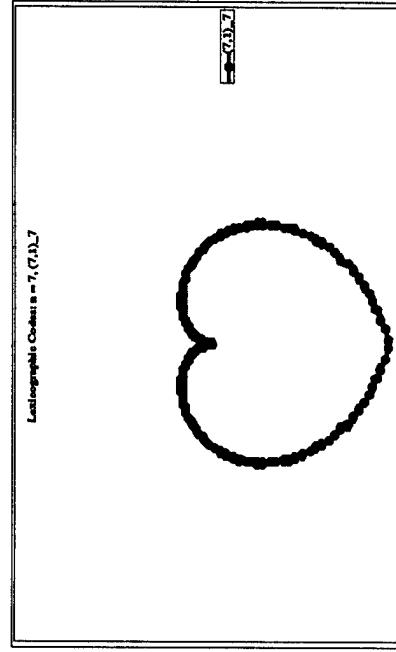


Fig. 81. Lexicographic Code, 7,0,8 (Appendix: Table 16).

## Appendix: Table of Contents

Table 1: Lexicographic-Greedy Code of Length  $n = 5, k = 2$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 4$ , with ordered basis:  $y_1=10000, y_2=01000, y_3=00100$ , etc.

Table 2: Lexicographic-Greedy Code of Length  $n = 5, k = 2$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 4$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 3: Gray-Greedy Code of Length  $n = 5, k = 2$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 4$ , with ordered basis:  $y_1=00001, y_2=00011, y_3=00110$ , etc.

Table 4: Standard Array for the Lexicographic-Greedy Code (7,4) of Table 7.

Table 5: Standard Array for the Lexicographic-Greedy Code (7,4) of Table 8.

Table 6: Standard Array for the Lexicographic-Greedy Code (7,4) of Table 9.

Table 7: Lexicographic-Greedy Code of Length  $n = 7, k = 4$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 16$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 8: Lexicographic-Greedy Code of Length  $n = 7, k = 4$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 16$ , with ordered basis:  $y_1=00001, y_2=00011, y_3=00110$ , etc.

Table 9: Lexicographic-Greedy Code of Length  $n = 7, k = 4$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 16$ , with ordered basis:  $y_1=00001, y_2=00011, y_3=00111$ , etc.

Table 10: Lexicographic-Greedy Code of Length  $n = 7, k = 6$  and Distance  $d = 2, t = (d-1)/2, M = 2^k = 64$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 11: Lexicographic-Greedy Code of Length  $n = 7, k = 4$  and Distance  $d = 3, t = (d-1)/2, M = 2^k = 16$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 12: Lexicographic-Greedy Code of Length  $n = 7, k = 3$  and Distance  $d = 4, t = (d-1)/2, M = 2^k = 8$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 13: Lexicographic-Greedy Code of Length  $n = 7, k = 1$  and Distance  $d = 5, t = (d-1)/2, M = 2^k = 2$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 14: Lexicographic-Greedy Code of Length  $n = 7, k = 1$  and Distance  $d = 6, t = (d-1)/2, M = 2^k = 2$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 15: Lexicographic-Greedy Code of Length  $n = 7, k = 1$  and Distance  $d = 7, t = (d-1)/2, M = 2^k = 2$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 16: Lexicographic-Greedy Code of Length  $n = 7, k = 0$  and Distance  $d = 8, t = (d-1)/2, M = 2^k = 1$ , with ordered basis:  $y_1=00001, y_2=00010, y_3=00100$ , etc.

Table 17: Greedy Code Algorithm

**Appendix: Table 1. Lexicographic-Greedy Code of Length  $n = 5$ ,  $k = 2$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 4$ , with ordered basis:**

$$\begin{aligned}
 \mathbf{y}_1 &= 10000 \\
 \mathbf{y}_2 &= 01000 \\
 \mathbf{y}_3 &= 00100 \\
 \mathbf{y}_4 &= 00010 \\
 \mathbf{y}_5 &= 00001
 \end{aligned}$$

$y_1 = \{0, 0, 0, 0, 0\}$	0	•
$y_2 = \{1, 0, 0, 0, 0\}$	1	
$y_3 = \{0, 1, 0, 0, 0\}$	2	
$\{1, 1, 0, 0, 0\}$	3	
$y_4 = \{0, 0, 1, 0, 0\}$	3	
$\{1, 0, 1, 0, 0\}$	2	
$\{0, 1, 1, 0, 0\}$	1	
$\{1, 1, 1, 0, 0\}$	0	•
$y_5 = \{0, 0, 0, 1, 0\}$	4	
$\{1, 0, 0, 1, 0\}$	5	
$\{0, 1, 0, 1, 0\}$	6	
$\{1, 1, 0, 1, 0\}$	7	
$\{0, 0, 1, 1, 0\}$	7	
$\{1, 0, 1, 1, 0\}$	6	
$\{0, 1, 1, 1, 0\}$	5	
$\{1, 1, 1, 1, 0\}$	4	
$y_5 = \{0, 0, 0, 0, 1\}$	5	
$\{1, 0, 0, 0, 1\}$	4	
$\{0, 1, 0, 0, 1\}$	7	
$\{1, 1, 0, 0, 1\}$	6	
$\{0, 0, 1, 0, 1\}$	6	
$\{1, 0, 1, 0, 1\}$	7	
$\{0, 1, 1, 0, 1\}$	4	
$\{1, 1, 1, 0, 1\}$	5	
$\{0, 0, 0, 1, 1\}$	1	
$\{1, 0, 0, 1, 1\}$	0	•
$\{0, 1, 0, 1, 1\}$	3	
$\{1, 1, 0, 1, 1\}$	2	
$\{0, 0, 1, 1, 1\}$	2	
$\{1, 0, 1, 1, 1\}$	3	
$\{0, 1, 1, 1, 1\}$	0	•
$\{1, 1, 1, 1, 1\}$	1	

**Appendix: Table 2. Lexicographic-Greedy Code of Length  $n = 5$ ,  $k = 2$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 4$ , with ordered basis:**

$$\begin{aligned}
 \mathbf{y}_1 &= 00001 \\
 \mathbf{y}_2 &= 00010 \\
 \mathbf{y}_3 &= 00100 \\
 \mathbf{y}_4 &= 01000 \\
 \mathbf{y}_5 &= 10000
 \end{aligned}$$

Lexicographic order of $F_2^n$	g values	g values as vectors	Lexicographic code
{0, 0, 0, 0, 0}	0	000	•
y1={0, 0, 0, 0, 1}	1	001	
y2={0, 0, 0, 1, 0}	2	010	
{0, 0, 0, 1, 1}	3	011	
y3={0, 0, 1, 0, 0}	3	011	
{0, 0, 1, 0, 1}	2	010	
{0, 0, 1, 1, 0}	1	001	
{0, 0, 1, 1, 1}	0	000	
y4={0, 1, 0, 0, 0}	4	100	•
{0, 1, 0, 0, 1}	5	101	
{0, 1, 0, 1, 0}	6	110	
{0, 1, 0, 1, 1}	7	111	
{0, 1, 1, 0, 0}	7	111	
{0, 1, 1, 0, 1}	6	110	
{0, 1, 1, 1, 0}	5	101	
{0, 1, 1, 1, 1}	4	100	
y5={1, 0, 0, 0, 0}	5	101	•
{1, 0, 0, 0, 1}	4	100	
{1, 0, 0, 1, 0}	7	111	
{1, 0, 0, 1, 1}	6	110	
{1, 0, 1, 0, 0}	6	110	
{1, 0, 1, 0, 1}	7	111	
{1, 0, 1, 1, 0}	4	100	
{1, 0, 1, 1, 1}	5	101	
{1, 1, 0, 0, 0}	1	001	
{1, 1, 0, 0, 1}	0	000	
{1, 1, 0, 1, 0}	3	011	
{1, 1, 0, 1, 1}	2	010	
{1, 1, 1, 0, 0}	2	010	
{1, 1, 1, 0, 1}	3	011	
{1, 1, 1, 1, 0}	0	000	
{1, 1, 1, 1, 1}	1	001	

**Appendix: Table 3. Gray-Greedy Code<sup>83</sup> of Length  $n = 5$ ,  $k = 2$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 4$ , with ordered basis:**

$$\begin{aligned}
 y_1 &= 00001 \\
 y_2 &= 00011 \\
 y_3 &= 00110 \\
 y_4 &= 01100 \\
 y_5 &= 11000
 \end{aligned}$$

Lexicographic order of $F_2^n$	g values	g values as vectors	Gray-Greedy code
{0, 0, 0, 0, 0}	0	000	.
y1={0, 0, 0, 0, 1}	1	001	.
y2={0, 0, 0, 1, 1}	2	010	.
{0, 0, 0, 1, 0}	3	011	.
y3={0, 0, 1, 1, 0}	1	001	.
{0, 0, 1, 1, 1}	0	000	.
{0, 0, 1, 0, 1}	3	011	.
{0, 0, 1, 0, 0}	2	010	.
y4={0, 1, 1, 0, 0}	4	100	.
{0, 1, 1, 0, 1}	5	101	.
{0, 1, 1, 1, 1}	6	110	.
{0, 1, 1, 1, 0}	7	111	.
{0, 1, 0, 1, 0}	5	101	.
{0, 1, 0, 1, 1}	4	100	.
{0, 1, 0, 0, 1}	7	111	.
{0, 1, 0, 0, 0}	6	110	.
y5={1, 1, 0, 0, 0}	1	001	.
{1, 1, 0, 0, 1}	0	000	.
{1, 1, 0, 1, 1}	3	011	.
{1, 1, 0, 1, 0}	2	010	.
{1, 1, 1, 1, 0}	0	000	.
{1, 1, 1, 1, 1}	1	001	.
{1, 1, 1, 0, 1}	2	010	.
{1, 1, 1, 0, 0}	3	011	.
{1, 0, 1, 0, 0}	5	101	.
{1, 0, 1, 0, 1}	4	100	.
{1, 0, 1, 1, 1}	7	111	.
{1, 0, 1, 1, 0}	6	110	.
{1, 0, 0, 1, 0}	4	100	.
{1, 0, 0, 1, 1}	5	101	.
{1, 0, 0, 0, 1}	6	110	.
{1, 0, 0, 0, 0}	7	111	.

<sup>83</sup> After Brualdi, R.A. & Pless, V.S., Greedy codes. *J. of Combinatorial Theory, Series A* 64, 10-30, 1993.

Appendix: Table 4.

Standard Array for the Lexicographic-Greedy Code (7,4) of Table 7. Basis vectors are shaded.

g values of the z vectors of the Lexicographic Code (7,4) of Table 7

## Appendix: Table 5.

### Standard Array for the Lexicographic-Greedy Code (7,4) of Table 8

Coast Leaders	00000000	00001111	00110001	00111110	01010010	01011011	10010100	10011000	10101011	11000001	11001100	11111111
00000100	00000100	00001111	00110001	00111110	01010010	01011011	10010100	10011000	10101011	11000001	11001100	11111110
00000110	00000110	00011101	00111111	00111111	01011010	01011010	01010101	01010101	01010101	11001111	11001111	11111110
10000000	10000000	10001111	10011001	10110011	01000000	01011111	01010001	01011111	01011111	11000010	11001100	11111011
01000000	01000000	01001111	01110001	01111110	00010000	01101101	11001001	00001011	00010010	00100001	01001111	01111011
00001000	00001000	00011111	00011001	00011110	00010010	00011011	00010011	00011011	00011011	11000001	10000010	10111111
00100000	00100000	00010111	00010001	00010001	00100010	00100010	01110101	01110101	01110101	11010001	11011110	11101011

*g* values of the *z* vectors of the Lexicographic Code (7,4) of Table 8

Appendix: Table 6.

Standard Array for the Lexicographic-Greedy Code (7,4) of Table 9. Basis vectors are shaded.

Coset Leaders	0000000	0011001	0011110	0101010	0101101	0110011	0110100	1001100	1010010	1010101	1100001	1100110
	0000010	0000110	0011000	0011011	0101100	0110010	0110101	1001101	1010011	1010110	1100000	1100111
	0000110	0001000	0011101	0011010	0101001	0110111	0110000	1001111	1010110	1010001	1100010	1111000
	0001000	0001100	0011101	0011010	0101001	0110111	0110000	1001111	1010000	1010111	1100011	1111011
	0001010	0001101	0011100	0011000	0101000	0110111	0110001	1001101	1010000	1010110	1100010	1111010
	0001100	0001100	0011001	0010010	0100101	0111011	0111100	1000011	1000100	1011010	1101101	1110000
	0001110	0001110	0011001	0011110	0111010	0110011	0100011	0100100	1011000	1000101	1110001	1110110
	0010000	0010000	0010111	0011001	0101100	0110101	0110011	0100100	0100100	0101101	0110001	0100110
	0010010	0010010	0010111	0011001	0101100	0110101	0110011	0100100	0100100	0101101	0110001	0100110
	0010110	0010110	0011001	0011110	0111010	0110101	0110011	0001100	0010010	0010101	0100001	0100110
	0011000	0011000	0011110	0011110	0111110	0110101	0001101	0010100	1110010	1110101	1000110	1011000
	0011100	0011100	0011110	0011110	0111110	0110101	0001101	0010100	1110100	1110101	1000110	1011000
	0100000	0100000	0100111	0111001	0111110	0110101	0001101	0010100	1110010	1110101	1000001	1011000
	0100010	0100010	0100111	0111001	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0100110	0100110	0101000	0111001	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0101000	0101000	0101110	0111001	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0101100	0101100	0101110	0111001	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0110000	0110000	0110111	0111001	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0110110	0110110	0111001	0111110	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0111000	0111000	0111110	0111110	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	0111100	0111100	0111110	0111110	0111110	0110101	0001101	0010100	1110100	1110101	1000001	1011000
	1011000	1011000	1011110	1011110	1011110	1010101	0001101	0010100	1110100	1110101	1000001	1011000
	1011100	1011100	1011110	1011110	1011110	1010101	0001101	0010100	1110100	1110101	1000001	1011000
	1110000	1110000	1110110	1110110	1110110	1110101	0001101	0010100	1110100	1110101	1000001	1011000
	1110110	1110110	1110110	1110110	1110110	1110101	0001101	0010100	1110100	1110101	1000001	1011000
	1111000	1111000	1111000	1111000	1111000	1110101	0001101	0010100	1110100	1110101	1000001	1011000
	1111100	1111100	1111100	1111100	1111100	1110101	0001101	0010100	1110100	1110101	1000001	1011000
	1111110	1111110	1111110	1111110	1111110	1110101	0001101	0010100	1110100	1110101	1000001	1011000
	1111111	1111111	1111111	1111111	1111111	1110101	0001101	0010100	1110100	1110101	1000001	1011000

 $g^*$  values of the  $z$  vectors of the Lexicographic Code (7,4) of Table 9

0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7

**Appendix: Table 7. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 4$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 64$ , with ordered basis:**

$y_1 =$	0000001
$y_2 =$	0000010
$y_3 =$	0000100
$y_4 =$	0001000
$y_5 =$	0010000
$y_6 =$	0100000
$y_7 =$	1000000

Lexicographic order of $F_2^7$	g values	Lexicographic Code
{0, 0, 0, 0, 0, 0, 0}	0	•
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 0}	2	
{0, 0, 0, 0, 0, 1, 1}	3	
y3={0, 0, 0, 0, 1, 0, 0}	3	
{0, 0, 0, 0, 1, 0, 1}	2	
{0, 0, 0, 0, 1, 1, 0}	1	
{0, 0, 0, 0, 1, 1, 1}	0	
y4={0, 0, 0, 1, 0, 0, 0}	4	
{0, 0, 0, 1, 0, 0, 1}	5	
{0, 0, 0, 1, 0, 1, 0}	6	
{0, 0, 0, 1, 0, 1, 1}	7	
{0, 0, 0, 1, 1, 0, 0}	7	
{0, 0, 0, 1, 1, 0, 1}	6	
{0, 0, 0, 1, 1, 1, 0}	5	
{0, 0, 0, 1, 1, 1, 1}	4	
y5={0, 0, 1, 0, 0, 0, 0}	5	
{0, 0, 1, 0, 0, 0, 1}	4	
{0, 0, 1, 0, 0, 1, 0}	7	
{0, 0, 1, 0, 0, 1, 1}	6	
{0, 0, 1, 0, 1, 0, 0}	6	
{0, 0, 1, 0, 1, 0, 1}	7	
{0, 0, 1, 0, 1, 1, 0}	4	
{0, 0, 1, 0, 1, 1, 1}	5	
{0, 0, 1, 1, 0, 0, 0}	1	
{0, 0, 1, 1, 0, 0, 1}	0	
{0, 0, 1, 1, 0, 1, 0}	3	
{0, 0, 1, 1, 0, 1, 1}	2	
{0, 0, 1, 1, 1, 0, 0}	2	
{0, 0, 1, 1, 1, 0, 1}	3	
{0, 0, 1, 1, 1, 1, 0}	0	
{0, 0, 1, 1, 1, 1, 1}	1	
y6={0, 1, 0, 0, 0, 0, 0}	6	
{0, 1, 0, 0, 0, 0, 1}	7	
{0, 1, 0, 0, 0, 1, 0}	4	
{0, 1, 0, 0, 0, 1, 1}	5	
{0, 1, 0, 0, 1, 0, 0}	5	
{0, 1, 0, 0, 1, 0, 1}	4	
{0, 1, 0, 0, 1, 1, 0}	7	

{0, 1, 0, 0, 1, 1, 1}	6	•
{0, 1, 0, 1, 0, 0, 0}	2	
{0, 1, 0, 1, 0, 0, 1}	3	
{0, 1, 0, 1, 0, 1, 0}	0	
{0, 1, 0, 1, 0, 1, 1}	1	
{0, 1, 0, 1, 1, 0, 0}	1	
{0, 1, 0, 1, 1, 0, 1}	0	
{0, 1, 0, 1, 1, 1, 0}	3	
{0, 1, 0, 1, 1, 1, 1}	2	
{0, 1, 1, 0, 0, 0, 0}	3	
{0, 1, 1, 0, 0, 0, 1}	2	
{0, 1, 1, 0, 0, 1, 0}	1	
{0, 1, 1, 0, 0, 1, 1}	0	
{0, 1, 1, 0, 1, 0, 0}	0	
{0, 1, 1, 0, 1, 0, 1}	1	
{0, 1, 1, 0, 1, 1, 0}	2	
{0, 1, 1, 0, 1, 1, 1}	3	
{0, 1, 1, 1, 0, 0, 0}	2	
{0, 1, 1, 1, 0, 0, 1}	7	
{0, 1, 1, 1, 0, 1, 0}	6	
{0, 1, 1, 1, 0, 1, 1}	5	
{0, 1, 1, 1, 1, 0, 0}	4	
{0, 1, 1, 1, 1, 0, 1}	4	
{0, 1, 1, 1, 1, 1, 0}	5	
{0, 1, 1, 1, 1, 1, 1}	6	
{0, 1, 1, 1, 1, 1, 1}	7	
{0, 1, 1, 1, 1, 1, 1}	7	
{0, 1, 1, 1, 1, 1, 1}	6	
{0, 1, 1, 1, 1, 1, 1}	5	
{0, 1, 1, 1, 1, 1, 1}	4	
{0, 1, 1, 1, 1, 1, 1}	3	
{0, 1, 1, 1, 1, 1, 1}	2	
{0, 1, 1, 1, 1, 1, 1}	1	
{0, 1, 1, 1, 1, 1, 1}	0	
{1, 0, 0, 0, 0, 0, 0}	7	•
{1, 0, 0, 0, 0, 0, 1}	6	
{1, 0, 0, 0, 0, 1, 0}	5	
{1, 0, 0, 0, 0, 1, 1}	4	
{1, 0, 0, 0, 1, 0, 0}	4	
{1, 0, 0, 0, 1, 0, 1}	5	
{1, 0, 0, 0, 1, 1, 0}	6	
{1, 0, 0, 0, 1, 1, 1}	7	
{1, 0, 0, 1, 0, 0, 0}	3	
{1, 0, 0, 1, 0, 0, 1}	2	
{1, 0, 0, 1, 0, 1, 0}	1	
{1, 0, 0, 1, 0, 1, 1}	0	
{1, 0, 0, 1, 1, 0, 0}	0	
{1, 0, 0, 1, 1, 0, 1}	1	
{1, 0, 0, 1, 1, 1, 0}	2	
{1, 0, 0, 1, 1, 1, 1}	3	
{1, 0, 1, 0, 0, 0, 0}	2	
{1, 0, 1, 0, 0, 0, 1}	3	
{1, 0, 1, 0, 0, 1, 0}	0	
{1, 0, 1, 0, 0, 1, 1}	1	
{1, 0, 1, 0, 1, 0, 0}	1	
{1, 0, 1, 0, 1, 0, 1}	0	
{1, 0, 1, 0, 1, 1, 0}	3	
{1, 0, 1, 0, 1, 1, 1}	2	
{1, 0, 1, 1, 0, 0, 0}	6	
{1, 0, 1, 1, 0, 0, 1}	7	
{1, 0, 1, 1, 0, 1, 0}	4	
{1, 0, 1, 1, 0, 1, 1}	5	
{1, 0, 1, 1, 1, 0, 0}	5	

{1, 0, 1, 1, 1, 0, 1}	4
{1, 0, 1, 1, 1, 1, 0}	7
{1, 0, 1, 1, 1, 1, 1}	6
{1, 1, 0, 0, 0, 0, 0}	1
{1, 1, 0, 0, 0, 0, 1}	0
{1, 1, 0, 0, 0, 1, 0}	3
{1, 1, 0, 0, 0, 1, 1}	2
{1, 1, 0, 0, 1, 0, 0}	2
{1, 1, 0, 0, 1, 0, 1}	3
{1, 1, 0, 0, 1, 1, 0}	0
{1, 1, 0, 0, 1, 1, 1}	1
{1, 1, 0, 1, 0, 0, 0}	5
{1, 1, 0, 1, 0, 0, 1}	4
{1, 1, 0, 1, 0, 1, 0}	7
{1, 1, 0, 1, 0, 1, 1}	6
{1, 1, 0, 1, 1, 0, 0}	6
{1, 1, 0, 1, 1, 0, 1}	7
{1, 1, 0, 1, 1, 1, 0}	4
{1, 1, 0, 1, 1, 1, 1}	5
{1, 1, 0, 1, 1, 1, 1}	4
{1, 1, 0, 1, 1, 1, 1}	5
{1, 1, 0, 1, 1, 1, 1}	4
{1, 1, 1, 0, 0, 0, 0}	5
{1, 1, 1, 0, 0, 0, 1}	6
{1, 1, 1, 0, 0, 1, 0}	7
{1, 1, 1, 0, 0, 1, 1}	6
{1, 1, 1, 0, 1, 0, 0}	6
{1, 1, 1, 0, 1, 0, 1}	7
{1, 1, 1, 0, 1, 1, 0}	4
{1, 1, 1, 0, 1, 1, 1}	5
{1, 1, 1, 1, 0, 0, 0}	4
{1, 1, 1, 1, 0, 0, 1}	5
{1, 1, 1, 1, 0, 1, 0}	6
{1, 1, 1, 1, 0, 1, 1}	7
{1, 1, 1, 1, 1, 0, 0}	7
{1, 1, 1, 1, 1, 0, 1}	8
{1, 1, 1, 1, 1, 1, 0}	9
{1, 1, 1, 1, 1, 1, 1}	10

**Appendix: Table 8. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 4$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 64$ , with ordered basis:**

$y_1$ =	0000001
$y_2$ =	0000011
$y_3$ =	0000110
$y_4$ =	0001100
$y_5$ =	0011000
$y_6$ =	0110000
$y_7$ =	1100000

Lexicographic order of $F_2^7$	g values	Lexicographic Code
{0, 0, 0, 0, 0, 0, 0}	0	•
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 1}	2	
{0, 0, 0, 0, 1, 0, 0}	3	
y3={0, 0, 0, 0, 1, 1, 0}	1	
{0, 0, 0, 1, 1, 1, 1}	0	•
{0, 0, 0, 0, 1, 0, 1}	3	
{0, 0, 0, 0, 1, 0, 0}	2	
y4={0, 0, 0, 1, 1, 0, 0}	4	
{0, 0, 0, 1, 1, 0, 1}	5	
{0, 0, 0, 1, 1, 1, 1}	6	
{0, 0, 0, 1, 1, 1, 0}	7	
{0, 0, 0, 1, 0, 1, 0}	5	
{0, 0, 0, 1, 0, 1, 1}	4	
{0, 0, 0, 1, 0, 0, 1}	7	
{0, 0, 0, 1, 0, 0, 0}	6	
y5={0, 0, 1, 1, 0, 0, 0}	1	•
{0, 0, 1, 1, 0, 0, 1}	0	
{0, 0, 1, 1, 0, 1, 1}	3	
{0, 0, 1, 1, 0, 1, 0}	2	
{0, 0, 1, 1, 1, 1, 0}	0	•
{0, 0, 1, 1, 1, 1, 1}	1	
{0, 0, 1, 1, 1, 1, 0}	2	
{0, 0, 1, 1, 1, 0, 0}	3	
{0, 0, 1, 0, 1, 0, 0}	5	
{0, 0, 1, 0, 1, 0, 1}	4	
{0, 0, 1, 0, 1, 1, 1}	7	
{0, 0, 1, 0, 1, 1, 0}	6	
{0, 0, 1, 0, 0, 1, 1}	4	
{0, 0, 1, 0, 0, 1, 0}	5	
{0, 0, 1, 0, 0, 0, 1}	6	
{0, 0, 1, 0, 0, 0, 0}	7	
y6={0, 1, 1, 0, 0, 0, 0}	2	•
{0, 1, 1, 0, 0, 0, 1}	3	
{0, 1, 1, 0, 0, 1, 1}	0	
{0, 1, 1, 0, 0, 1, 0}	1	
{0, 1, 1, 0, 1, 1, 0}	3	
{0, 1, 1, 0, 1, 1, 1}	2	
{0, 1, 1, 0, 1, 0, 1}	1	
{0, 1, 1, 0, 1, 0, 0}	0	•

{0, 1, 1, 1, 1, 0, 0}	6
{0, 1, 1, 1, 1, 0, 1}	7
{0, 1, 1, 1, 1, 1, 1}	4
{0, 1, 1, 1, 1, 1, 0}	5
{0, 1, 1, 1, 1, 0, 1}	7
{0, 1, 1, 1, 1, 0, 0}	7
{0, 1, 1, 1, 0, 1, 1}	6
{0, 1, 1, 1, 0, 0, 1}	5
{0, 1, 1, 1, 0, 0, 0}	4
{0, 1, 0, 1, 0, 0, 0}	3
{0, 1, 0, 1, 0, 0, 1}	2
{0, 1, 0, 1, 0, 1, 1}	1
{0, 1, 0, 1, 0, 1, 0}	0
{0, 1, 0, 1, 1, 1, 0}	2
{0, 1, 0, 1, 1, 1, 1}	3
{0, 1, 0, 1, 1, 0, 1}	0
{0, 1, 0, 1, 1, 0, 0}	1
{0, 1, 0, 0, 1, 0, 0}	7
{0, 1, 0, 0, 1, 0, 1}	6
{0, 1, 0, 0, 1, 1, 1}	5
{0, 1, 0, 0, 1, 1, 0}	4
{0, 1, 0, 0, 0, 1, 0}	6
{0, 1, 0, 0, 0, 1, 1}	7
{0, 1, 0, 0, 0, 0, 1}	4
{0, 1, 0, 0, 0, 0, 0}	5
y7={1, 1, 0, 0, 0, 0, 0}	1
{1, 1, 0, 0, 0, 0, 1}	0
{1, 1, 0, 0, 0, 1, 1}	3
{1, 1, 0, 0, 0, 1, 0}	2
{1, 1, 0, 0, 1, 1, 0}	0
{1, 1, 0, 0, 1, 1, 1}	1
{1, 1, 0, 0, 1, 0, 1}	2
{1, 1, 0, 0, 1, 0, 0}	3
{1, 1, 0, 1, 1, 0, 0}	5
{1, 1, 0, 1, 1, 0, 1}	4
{1, 1, 0, 1, 1, 1, 1}	7
{1, 1, 0, 1, 1, 1, 0}	6
{1, 1, 0, 1, 0, 1, 0}	4
{1, 1, 0, 1, 0, 1, 1}	5
{1, 1, 0, 1, 0, 0, 1}	6
{1, 1, 0, 1, 0, 0, 0}	7
{1, 1, 1, 1, 0, 0, 0}	0
{1, 1, 1, 1, 0, 0, 1}	1
{1, 1, 1, 1, 0, 1, 1}	2
{1, 1, 1, 1, 0, 1, 0}	3
{1, 1, 1, 1, 1, 1, 0}	1
{1, 1, 1, 1, 1, 1, 1}	0
{1, 1, 1, 1, 1, 0, 0}	3
{1, 1, 1, 1, 1, 0, 1}	2
{1, 1, 1, 1, 0, 1, 0}	4
{1, 1, 1, 1, 0, 1, 1}	5
{1, 1, 1, 1, 0, 0, 1}	6
{1, 1, 1, 1, 0, 0, 0}	7
{1, 1, 1, 0, 0, 0, 1}	5
{1, 1, 1, 1, 0, 0, 1}	4
{1, 1, 1, 1, 0, 1, 0}	5
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 0, 0, 0}	7
{1, 1, 1, 0, 0, 0, 1}	5
{1, 1, 1, 1, 0, 0, 1}	4
{1, 1, 1, 1, 0, 1, 0}	5
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 0, 0, 0}	7
{1, 1, 1, 0, 0, 0, 1}	5
{1, 1, 1, 1, 0, 0, 1}	4

{1, 1, 1, 0, 0, 0, 1}	7
{1, 1, 1, 0, 0, 0, 0}	6
{1, 0, 1, 0, 0, 0, 0}	3
{1, 0, 1, 0, 0, 0, 1}	2
{1, 0, 1, 0, 0, 1, 1}	1
{1, 0, 1, 0, 0, 1, 0}	0
{1, 0, 1, 0, 1, 1, 0}	2
{1, 0, 1, 0, 1, 1, 1}	3
{1, 0, 1, 0, 1, 1, 1}	3
{1, 0, 1, 0, 1, 0, 1}	0
{1, 0, 1, 0, 1, 0, 0}	1
{1, 0, 1, 1, 1, 0, 0}	7
{1, 0, 1, 1, 1, 0, 1}	6
{1, 0, 1, 1, 1, 1, 1}	5
{1, 0, 1, 1, 1, 1, 1}	5
{1, 0, 1, 1, 1, 1, 0}	4
{1, 0, 1, 1, 0, 1, 0}	6
{1, 0, 1, 1, 0, 1, 1}	7
{1, 0, 1, 1, 0, 0, 1}	4
{1, 0, 1, 1, 0, 0, 0}	5
{1, 0, 0, 1, 0, 0, 0}	2
{1, 0, 0, 1, 0, 0, 1}	3
{1, 0, 0, 1, 0, 1, 1}	0
{1, 0, 0, 1, 0, 1, 0}	1
{1, 0, 0, 1, 1, 1, 0}	3
{1, 0, 0, 1, 1, 1, 1}	2
{1, 0, 0, 1, 1, 1, 0}	1
{1, 0, 0, 1, 1, 0, 0}	0
{1, 0, 0, 0, 1, 0, 0}	6
{1, 0, 0, 0, 1, 0, 1}	7
{1, 0, 0, 0, 1, 1, 1}	4
{1, 0, 0, 0, 1, 1, 0}	5
{1, 0, 0, 0, 0, 1, 0}	7
{1, 0, 0, 0, 0, 1, 1}	6
{1, 0, 0, 0, 0, 0, 1}	5
{1, 0, 0, 0, 0, 0, 0}	4

**Appendix: Table 9. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 4$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 64$ , with ordered basis:**

$y_1 =$	0000001
$y_2 =$	0000011
$y_3 =$	0000111
$y_4 =$	0001111
$y_5 =$	0011111
$y_6 =$	0111111
$y_7 =$	1111111

Lexicographic order of $F_2^7$	g values	Lexicographic Code
{0, 0, 0, 0, 0, 0, 0}	0	•
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 1}	2	
{0, 0, 0, 0, 1, 0, 0}	3	
y3={0, 0, 0, 0, 1, 1, 1}	0	•
{0, 0, 0, 1, 1, 0, 0}	1	
{0, 0, 0, 1, 0, 0, 0}	2	
{0, 0, 0, 1, 0, 0, 1}	3	
y4={0, 0, 0, 1, 1, 1, 1}	4	
{0, 0, 0, 1, 1, 1, 0}	5	
{0, 0, 0, 1, 1, 0, 0}	6	
{0, 0, 0, 1, 1, 0, 1}	7	
{0, 0, 0, 1, 0, 0, 0}	4	
{0, 0, 0, 1, 0, 0, 1}	5	
{0, 0, 0, 1, 0, 1, 1}	6	
{0, 0, 0, 1, 0, 1, 0}	7	
y5={0, 0, 1, 1, 1, 1, 1}	1	
{0, 0, 1, 1, 1, 1, 0}	0	•
{0, 0, 1, 1, 1, 0, 0}	3	
{0, 0, 1, 1, 1, 0, 1}	2	
{0, 0, 1, 1, 0, 0, 0}	1	
{0, 0, 1, 1, 0, 0, 1}	0	•
{0, 0, 1, 1, 0, 1, 1}	3	
{0, 0, 1, 1, 0, 1, 0}	2	
{0, 0, 1, 1, 0, 0, 0}	5	
{0, 0, 1, 0, 0, 0, 0}	4	
{0, 0, 1, 0, 0, 0, 1}	7	
{0, 0, 1, 0, 0, 1, 1}	6	
{0, 0, 1, 0, 0, 1, 0}	5	
{0, 0, 1, 0, 1, 1, 1}	5	
{0, 0, 1, 0, 1, 1, 0}	4	
{0, 0, 1, 0, 1, 0, 0}	7	
{0, 0, 1, 0, 1, 0, 1}	6	
y6={0, 1, 1, 1, 1, 1, 1}	6	
{0, 1, 1, 1, 1, 1, 0}	7	
{0, 1, 1, 1, 1, 0, 0}	4	
{0, 1, 1, 1, 1, 0, 1}	5	
{0, 1, 1, 1, 0, 0, 0}	6	
{0, 1, 1, 1, 0, 0, 1}	7	
{0, 1, 1, 1, 0, 1, 1}	4	

{0, 1, 1, 1, 0, 1, 0}	5
{0, 1, 1, 0, 0, 0, 0}	2
{0, 1, 1, 0, 0, 0, 1}	3
{0, 1, 1, 0, 0, 1, 1}	0
{0, 1, 1, 0, 0, 1, 0}	1
{0, 1, 1, 0, 1, 1, 1}	2
{0, 1, 1, 0, 1, 1, 0}	3
{0, 1, 1, 0, 1, 0, 0}	0
{0, 1, 1, 0, 1, 0, 1}	1
{0, 1, 0, 0, 0, 0, 0}	7
{0, 1, 0, 0, 0, 0, 1}	6
{0, 1, 0, 0, 0, 1, 1}	5
{0, 1, 0, 0, 0, 1, 0}	4
{0, 1, 0, 0, 1, 1, 1}	7
{0, 1, 0, 0, 1, 1, 0}	6
{0, 1, 0, 0, 1, 0, 0}	5
{0, 1, 0, 0, 1, 0, 1}	4
{0, 1, 0, 1, 1, 1, 1}	3
{0, 1, 0, 1, 1, 1, 0}	2
{0, 1, 0, 1, 1, 0, 0}	1
{0, 1, 0, 1, 1, 0, 1}	0
{0, 1, 0, 1, 0, 0, 0}	3
{0, 1, 0, 1, 0, 0, 1}	2
{0, 1, 0, 1, 0, 1, 1}	1
{0, 1, 0, 1, 0, 1, 0}	0
{0, 1, 0, 1, 0, 0, 0}	0
{0, 1, 0, 1, 0, 0, 1}	0
{0, 1, 1, 1, 1, 1, 1}	0
{1, 1, 1, 1, 1, 1, 0}	1
{1, 1, 1, 1, 1, 0, 0}	2
{1, 1, 1, 1, 1, 0, 1}	3
{1, 1, 1, 1, 0, 0, 0}	0
{1, 1, 1, 1, 0, 0, 1}	1
{1, 1, 1, 1, 0, 1, 1}	2
{1, 1, 1, 1, 0, 1, 0}	3
{1, 1, 1, 1, 0, 0, 0}	4
{1, 1, 1, 1, 0, 0, 1}	5
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 0, 1, 0}	7
{1, 1, 1, 1, 0, 0, 1}	4
{1, 1, 1, 1, 0, 0, 0}	5
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 0, 1, 0}	7
{1, 1, 1, 1, 0, 0, 0}	4
{1, 1, 1, 1, 0, 0, 1}	5
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 0, 1, 0}	7
{1, 1, 1, 1, 0, 0, 0}	4
{1, 1, 1, 1, 0, 0, 1}	5
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 0, 1, 0}	7
{1, 1, 1, 1, 0, 0, 0}	5

{1, 1, 0, 1, 0, 0, 1}	4
{1, 1, 0, 1, 0, 1, 1}	7
{1, 1, 0, 1, 0, 1, 0}	6
{1, 0, 0, 0, 0, 0, 0}	6
{1, 0, 0, 0, 0, 0, 1}	7
{1, 0, 0, 0, 0, 1, 1}	4
{1, 0, 0, 0, 0, 1, 0}	5
{1, 0, 0, 0, 1, 1, 1}	6
{1, 0, 0, 0, 1, 1, 0}	7
{1, 0, 0, 0, 1, 0, 0}	4
{1, 0, 0, 0, 1, 0, 1}	5
{1, 0, 0, 1, 1, 1, 1}	2
{1, 0, 0, 1, 1, 1, 0}	3
{1, 0, 0, 1, 1, 0, 0}	0
{1, 0, 0, 1, 1, 0, 1}	1
{1, 0, 0, 1, 0, 0, 0}	2
{1, 0, 0, 1, 0, 0, 1}	3
{1, 0, 0, 1, 0, 1, 1}	0
{1, 0, 0, 1, 0, 1, 0}	1
{1, 0, 1, 1, 1, 1, 1}	7
{1, 0, 1, 1, 1, 1, 0}	6
{1, 0, 1, 1, 1, 0, 0}	5
{1, 0, 1, 1, 1, 0, 1}	4
{1, 0, 1, 1, 0, 0, 0}	7
{1, 0, 1, 1, 0, 0, 1}	6
{1, 0, 1, 1, 0, 1, 1}	5
{1, 0, 1, 1, 0, 1, 0}	4
{1, 0, 1, 0, 0, 0, 0}	3
{1, 0, 1, 0, 0, 0, 1}	2
{1, 0, 1, 0, 0, 1, 1}	1
{1, 0, 1, 0, 0, 1, 0}	0
{1, 0, 1, 0, 1, 1, 1}	3
{1, 0, 1, 0, 1, 1, 0}	2
{1, 0, 1, 0, 1, 0, 0}	1
{1, 0, 1, 0, 1, 0, 1}	0
{1, 0, 1, 0, 0, 0, 1}	1
{1, 0, 1, 0, 0, 1, 1}	0

**Appendix: Table 10. Lexicographic-Greedy Code of Length  $n = 7, k = 6$  and Distance  $d = 2, t = (d-1)/2, M = 2^k = 64$ , with ordered basis:**

$y_1 =$	0000001		
$y_2 =$	0000010		
$y_3 =$	0000100		
$y_4 =$	0001000		
$y_5 =$	0010000		
$y_6 =$	0100000		
$y_7 =$	1000000		
$\{0, 0, 0, 0, 0, 0, 0\}$	0	•	
$y_1 = \{0, 0, 0, 0, 0, 0, 1\}$	1		
$y_2 = \{0, 0, 0, 0, 0, 1, 0\}$	1		
$\{0, 0, 0, 0, 1, 1, 1\}$	0	•	
$y_3 = \{0, 0, 0, 0, 1, 0, 0\}$	1		
$\{0, 0, 0, 0, 1, 0, 1\}$	0	•	
$\{0, 0, 0, 0, 1, 1, 0\}$	0	•	
$\{0, 0, 0, 0, 1, 1, 1\}$	1		
$y_4 = \{0, 0, 0, 1, 0, 0, 0\}$	1		
$\{0, 0, 0, 1, 0, 0, 1\}$	0	•	
$\{0, 0, 0, 1, 0, 1, 0\}$	0	•	
$\{0, 0, 0, 1, 0, 1, 1\}$	1		
$\{0, 0, 0, 1, 1, 0, 0\}$	0	•	
$\{0, 0, 0, 1, 1, 0, 1\}$	1		
$\{0, 0, 0, 1, 1, 1, 0\}$	1		
$\{0, 0, 0, 1, 1, 1, 1\}$	0	•	
$y_5 = \{0, 0, 1, 0, 0, 0, 0\}$	1		
$\{0, 0, 1, 0, 0, 0, 1\}$	0	•	
$\{0, 0, 1, 0, 0, 1, 0\}$	0	•	
$\{0, 0, 1, 0, 0, 1, 1\}$	1		
$\{0, 0, 1, 0, 1, 0, 0\}$	0	•	
$\{0, 0, 1, 0, 1, 0, 1\}$	1		
$\{0, 0, 1, 0, 1, 1, 0\}$	1		
$\{0, 0, 1, 0, 1, 1, 1\}$	0	•	
$\{0, 0, 1, 1, 0, 0, 0\}$	0	•	
$\{0, 0, 1, 1, 0, 0, 1\}$	0	•	
$\{0, 0, 1, 1, 0, 1, 0\}$	1		
$\{0, 0, 1, 1, 0, 1, 1\}$	0	•	
$\{0, 0, 1, 1, 1, 0, 0\}$	0	•	
$\{0, 0, 1, 1, 1, 0, 1\}$	1		
$\{0, 0, 1, 1, 1, 1, 0\}$	1		
$y_6 = \{0, 1, 0, 0, 0, 0, 0\}$	1		
$\{0, 1, 0, 0, 0, 0, 1\}$	0	•	
$\{0, 1, 0, 0, 0, 1, 0\}$	0	•	
$\{0, 1, 0, 0, 0, 1, 1\}$	1		
$\{0, 1, 0, 0, 1, 0, 0\}$	0	•	
$\{0, 1, 0, 0, 1, 0, 1\}$	1		
$\{0, 1, 0, 0, 1, 1, 0\}$	1		
$\{0, 1, 0, 0, 1, 1, 1\}$	0	•	

{0, 1, 0, 1, 0, 0, 0}	0	•
{0, 1, 0, 1, 0, 0, 1}	1	
{0, 1, 0, 1, 0, 1, 0}	1	
{0, 1, 0, 1, 0, 1, 1}	0	•
{0, 1, 0, 1, 1, 0, 0}	1	
{0, 1, 0, 1, 1, 0, 1}	0	•
{0, 1, 0, 1, 1, 1, 0}	0	•
{0, 1, 0, 1, 1, 1, 1}	1	
{0, 1, 1, 0, 0, 0, 0}	0	•
{0, 1, 1, 0, 0, 0, 1}	1	
{0, 1, 1, 0, 0, 1, 0}	1	
{0, 1, 1, 0, 0, 1, 1}	0	•
{0, 1, 1, 0, 1, 0, 0}	1	
{0, 1, 1, 0, 1, 0, 1}	0	•
{0, 1, 1, 0, 1, 1, 0}	0	•
{0, 1, 1, 0, 1, 1, 1}	1	
{0, 1, 1, 1, 0, 0, 0}	0	•
{0, 1, 1, 1, 0, 0, 1}	1	
{0, 1, 1, 1, 0, 1, 0}	1	
{0, 1, 1, 1, 0, 1, 1}	0	•
{0, 1, 1, 1, 1, 0, 0}	1	
{0, 1, 1, 1, 1, 0, 1}	0	•
{0, 1, 1, 1, 1, 1, 0}	0	•
{0, 1, 1, 1, 1, 1, 1}	1	
{1, 0, 0, 0, 0, 0, 0}	1	
{1, 0, 0, 0, 0, 0, 1}	0	•
{1, 0, 0, 0, 0, 1, 0}	0	•
{1, 0, 0, 0, 0, 1, 1}	1	
{1, 0, 0, 0, 1, 0, 0}	0	•
{1, 0, 0, 0, 1, 0, 1}	1	
{1, 0, 0, 0, 1, 1, 0}	1	
{1, 0, 0, 0, 1, 1, 1}	0	•
{1, 0, 0, 1, 0, 0, 0}	0	•
{1, 0, 0, 1, 0, 0, 1}	1	
{1, 0, 0, 1, 0, 1, 0}	1	
{1, 0, 0, 1, 0, 1, 1}	0	•
{1, 0, 0, 1, 1, 0, 0}	1	
{1, 0, 0, 1, 1, 0, 1}	0	•
{1, 0, 0, 1, 1, 1, 0}	0	•
{1, 0, 0, 1, 1, 1, 1}	1	
{1, 0, 1, 0, 0, 0, 0}	0	•
{1, 0, 1, 0, 0, 0, 1}	1	
{1, 0, 1, 0, 0, 1, 0}	1	
{1, 0, 1, 0, 0, 1, 1}	0	•
{1, 0, 1, 0, 1, 0, 0}	1	
{1, 0, 1, 0, 1, 0, 1}	0	•
{1, 0, 1, 0, 1, 1, 0}	0	•
{1, 0, 1, 0, 1, 1, 1}	1	
{1, 0, 1, 1, 0, 0, 0}	1	
{1, 0, 1, 1, 0, 0, 1}	0	•
{1, 0, 1, 1, 0, 1, 0}	0	•
{1, 0, 1, 1, 0, 1, 1}	1	
{1, 0, 1, 1, 1, 0, 0}	0	•
{1, 0, 1, 1, 1, 0, 1}	1	
{1, 0, 1, 1, 1, 1, 0}	0	•
{1, 0, 1, 1, 1, 1, 1}	1	
{1, 0, 1, 1, 1, 1, 1}	1	

{1, 0, 1, 1, 1, 1, 0}	1	
{1, 0, 1, 1, 1, 1, 1}	0	•
{1, 1, 0, 0, 0, 0, 0}	0	
{1, 1, 0, 0, 0, 0, 1}	1	
{1, 1, 0, 0, 0, 1, 0}	1	
{1, 1, 0, 0, 0, 1, 1}	0	•
{1, 1, 0, 0, 1, 0, 0}	1	
{1, 1, 0, 0, 1, 0, 1}	0	•
{1, 1, 0, 0, 1, 1, 0}	0	•
{1, 1, 0, 0, 1, 1, 1}	1	
{1, 1, 0, 1, 0, 0, 0}	1	
{1, 1, 0, 1, 0, 0, 1}	0	•
{1, 1, 0, 1, 0, 1, 0}	0	•
{1, 1, 0, 1, 0, 1, 1}	0	•
{1, 1, 0, 1, 1, 0, 0}	0	•
{1, 1, 0, 1, 1, 0, 1}	1	
{1, 1, 0, 1, 1, 1, 0}	1	
{1, 1, 0, 1, 1, 1, 1}	0	•
{1, 1, 1, 0, 0, 0, 0}	1	
{1, 1, 1, 0, 0, 0, 1}	0	•
{1, 1, 1, 0, 0, 1, 0}	0	•
{1, 1, 1, 0, 0, 1, 1}	1	
{1, 1, 1, 0, 1, 0, 0}	0	•
{1, 1, 1, 0, 1, 0, 1}	1	
{1, 1, 1, 0, 1, 1, 0}	0	•
{1, 1, 1, 0, 1, 1, 1}	1	
{1, 1, 1, 1, 0, 0, 0}	0	•
{1, 1, 1, 1, 0, 0, 1}	1	
{1, 1, 1, 1, 0, 1, 0}	0	•
{1, 1, 1, 1, 0, 1, 1}	1	
{1, 1, 1, 1, 1, 0, 0}	0	•
{1, 1, 1, 1, 1, 0, 1}	1	
{1, 1, 1, 1, 1, 1, 0}	0	•
{1, 1, 1, 1, 1, 1, 1}	1	

**Appendix: Table 11. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 4$  and Distance  $d = 3$ ,  $t = (d-1)/2$ ,  $M = 2^k = 16$  with ordered basis:**

$y_1 =$	0000001
$y_2 =$	0000010
$y_3 =$	0000100
$y_4 =$	0001000
$y_5 =$	0010000
$y_6 =$	0100000
$y_7 =$	1000000

Lexicographic order of $F_2^7$	g values	Lexicographic Code
{0, 0, 0, 0, 0, 0, 0}	0	•
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 0}	2	
{0, 0, 0, 0, 0, 1, 1}	3	
y3={0, 0, 0, 0, 1, 0, 0}	3	
{0, 0, 0, 0, 1, 0, 1}	2	
{0, 0, 0, 0, 1, 1, 0}	1	
{0, 0, 0, 0, 1, 1, 1}	0	•
y4={0, 0, 0, 1, 0, 0, 0}	4	
{0, 0, 0, 1, 0, 0, 1}	5	
{0, 0, 0, 1, 0, 1, 0}	6	
{0, 0, 0, 1, 0, 1, 1}	7	
{0, 0, 0, 1, 1, 0, 0}	7	
{0, 0, 0, 1, 1, 0, 1}	6	
{0, 0, 0, 1, 1, 1, 0}	5	
{0, 0, 0, 1, 1, 1, 1}	4	
y5={0, 0, 1, 0, 0, 0, 0}	5	
{0, 0, 1, 0, 0, 0, 1}	4	
{0, 0, 1, 0, 0, 1, 0}	7	
{0, 0, 1, 0, 0, 1, 1}	6	
{0, 0, 1, 0, 1, 0, 0}	6	
{0, 0, 1, 0, 1, 0, 1}	7	
{0, 0, 1, 0, 1, 1, 0}	4	
{0, 0, 1, 0, 1, 1, 1}	5	
{0, 0, 1, 1, 0, 0, 0}	1	
{0, 0, 1, 1, 0, 0, 1}	0	•
{0, 0, 1, 1, 0, 1, 0}	3	
{0, 0, 1, 1, 0, 1, 1}	2	
{0, 0, 1, 1, 1, 0, 0}	2	
{0, 0, 1, 1, 1, 0, 1}	3	
{0, 0, 1, 1, 1, 1, 0}	0	•
{0, 0, 1, 1, 1, 1, 1}	1	
y6={0, 1, 0, 0, 0, 0, 0}	6	
{0, 1, 0, 0, 0, 0, 1}	7	
{0, 1, 0, 0, 0, 1, 0}	4	
{0, 1, 0, 0, 0, 1, 1}	5	
{0, 1, 0, 0, 1, 0, 0}	5	
{0, 1, 0, 0, 1, 0, 1}	4	
{0, 1, 0, 0, 1, 1, 0}	7	

{0, 1, 0, 0, 1, 1, 1}	6
{0, 1, 0, 1, 0, 0, 0}	2
{0, 1, 0, 1, 0, 0, 1}	3
{0, 1, 0, 1, 0, 1, 0}	0
{0, 1, 0, 1, 0, 1, 1}	1
{0, 1, 0, 1, 1, 0, 0}	1
{0, 1, 0, 1, 1, 0, 1}	0
{0, 1, 0, 1, 1, 1, 0}	3
{0, 1, 0, 1, 1, 1, 1}	2
{0, 1, 0, 1, 1, 1, 1}	2
{0, 1, 0, 1, 0, 0, 0}	3
{0, 1, 1, 0, 0, 0, 1}	2
{0, 1, 1, 0, 0, 1, 0}	1
{0, 1, 1, 0, 0, 1, 0}	1
{0, 1, 1, 0, 0, 1, 1}	0
{0, 1, 1, 0, 1, 0, 0}	0
{0, 1, 1, 0, 1, 0, 1}	1
{0, 1, 1, 0, 1, 1, 0}	2
{0, 1, 1, 0, 1, 1, 1}	3
{0, 1, 1, 1, 0, 0, 0}	2
{0, 1, 1, 1, 0, 0, 1}	1
{0, 1, 1, 1, 0, 1, 0}	1
{0, 1, 1, 1, 0, 1, 1}	0
{0, 1, 1, 1, 1, 0, 0}	0
{0, 1, 1, 1, 1, 0, 1}	1
{0, 1, 1, 1, 1, 1, 0}	2
{0, 1, 1, 1, 1, 1, 1}	3
{0, 1, 1, 1, 1, 0, 0}	7
{0, 1, 1, 1, 1, 0, 1}	6
{0, 1, 1, 1, 1, 1, 0}	5
{0, 1, 1, 1, 1, 1, 1}	4
{0, 1, 1, 1, 1, 0, 0}	4
{0, 1, 1, 1, 1, 0, 1}	5
{0, 1, 1, 1, 1, 1, 0}	6
{0, 1, 1, 1, 1, 1, 1}	7
y7={1, 0, 0, 0, 0, 0, 0}	7
{1, 0, 0, 0, 0, 0, 1}	6
{1, 0, 0, 0, 0, 1, 0}	5
{1, 0, 0, 0, 0, 1, 1}	4
{1, 0, 0, 0, 1, 0, 0}	4
{1, 0, 0, 0, 1, 0, 1}	5
{1, 0, 0, 0, 1, 1, 0}	6
{1, 0, 0, 0, 1, 1, 1}	7
{1, 0, 0, 1, 0, 0, 0}	3
{1, 0, 0, 1, 0, 0, 1}	2
{1, 0, 0, 1, 0, 1, 0}	1
{1, 0, 0, 1, 0, 1, 1}	0
{1, 0, 0, 1, 1, 0, 0}	0
{1, 0, 0, 1, 1, 0, 1}	1
{1, 0, 0, 1, 1, 1, 0}	2
{1, 0, 0, 1, 1, 1, 1}	3
{1, 0, 1, 0, 0, 0, 0}	2
{1, 0, 1, 0, 0, 0, 1}	3
{1, 0, 1, 0, 0, 1, 0}	0
{1, 0, 1, 0, 0, 1, 1}	1
{1, 0, 1, 0, 1, 0, 0}	1
{1, 0, 1, 0, 1, 0, 1}	0
{1, 0, 1, 0, 1, 1, 0}	3
{1, 0, 1, 0, 1, 1, 1}	2
{1, 0, 1, 1, 0, 0, 0}	6
{1, 0, 1, 1, 0, 0, 1}	7
{1, 0, 1, 1, 0, 1, 0}	4
{1, 0, 1, 1, 0, 1, 1}	5
{1, 0, 1, 1, 1, 0, 0}	5
{1, 0, 1, 1, 1, 0, 1}	5

{1, 0, 1, 1, 1, 0, 1}	4
{1, 0, 1, 1, 1, 1, 0}	7
{1, 0, 1, 1, 1, 1, 1}	6
{1, 1, 0, 0, 0, 0, 0}	1
{1, 1, 0, 0, 0, 0, 1}	0
{1, 1, 0, 0, 0, 1, 0}	3
{1, 1, 0, 0, 0, 1, 1}	2
{1, 1, 0, 0, 1, 0, 0}	2
{1, 1, 0, 0, 1, 0, 1}	3
{1, 1, 0, 0, 1, 1, 0}	0
{1, 1, 0, 0, 1, 1, 1}	1
{1, 1, 0, 1, 0, 0, 0}	5
{1, 1, 0, 1, 0, 0, 1}	4
{1, 1, 0, 1, 0, 1, 0}	7
{1, 1, 0, 1, 0, 1, 1}	6
{1, 1, 0, 1, 1, 0, 0}	6
{1, 1, 0, 1, 1, 0, 1}	7
{1, 1, 0, 1, 1, 1, 0}	4
{1, 1, 0, 1, 1, 1, 1}	5
{1, 1, 1, 0, 0, 0, 0}	4
{1, 1, 1, 0, 0, 0, 1}	5
{1, 1, 1, 0, 0, 1, 0}	6
{1, 1, 1, 0, 0, 1, 1}	7
{1, 1, 1, 0, 1, 0, 0}	7
{1, 1, 1, 0, 1, 0, 1}	6
{1, 1, 1, 0, 1, 1, 0}	5
{1, 1, 1, 0, 1, 1, 1}	4
{1, 1, 1, 1, 0, 0, 0}	5
{1, 1, 1, 1, 0, 0, 1}	6
{1, 1, 1, 1, 0, 1, 0}	7
{1, 1, 1, 1, 0, 1, 1}	6
{1, 1, 1, 1, 1, 0, 0}	7
{1, 1, 1, 1, 1, 0, 1}	6
{1, 1, 1, 1, 1, 1, 0}	5
{1, 1, 1, 1, 1, 1, 1}	4
{1, 1, 1, 1, 1, 1, 1, 0}	0
{1, 1, 1, 1, 1, 1, 1, 1}	1
{1, 1, 1, 1, 1, 0, 0, 1}	2
{1, 1, 1, 1, 1, 0, 1, 0}	3
{1, 1, 1, 1, 1, 1, 0, 0}	3
{1, 1, 1, 1, 1, 1, 0, 1}	2
{1, 1, 1, 1, 1, 1, 1, 0}	1
{1, 1, 1, 1, 1, 1, 1, 1}	0

•

•

•

•

**Appendix: Table 12. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 3$  and Distance  $d = 4$ ,  $t = (d-1)/2$ ,  $M = 2^k = 8$ , with ordered basis:**

$y_1 =$	0000001
$y_2 =$	0000010
$y_3 =$	0000100
$y_4 =$	0001000
$y_5 =$	0010000
$y_6 =$	0100000
$y_7 =$	1000000

Lexicographic order of $F_3^7$	g values	g values as vectors	Lexicographic code
{0, 0, 0, 0, 0, 0, 0}	0	0000	•
y1={0, 0, 0, 0, 0, 0, 1}	1	0001	
y2={0, 0, 0, 0, 0, 1, 0}	2	0010	
{0, 0, 0, 0, 0, 1, 1}	3	0011	
y3={0, 0, 0, 0, 1, 0, 0}	4	0100	
{0, 0, 0, 0, 1, 0, 1}	5	0101	
{0, 0, 0, 0, 1, 1, 0}	6	0110	
{0, 0, 0, 0, 1, 1, 1}	7	0111	
y4={0, 0, 0, 1, 0, 0, 0}	7	0111	
{0, 0, 0, 1, 0, 0, 1}	6	0110	
{0, 0, 0, 1, 0, 1, 0}	5	0101	
{0, 0, 0, 1, 0, 1, 1}	4	0100	
{0, 0, 0, 1, 1, 0, 0}	3	0011	
{0, 0, 0, 1, 1, 0, 1}	2	0010	
{0, 0, 0, 1, 1, 1, 0}	1	0001	
{0, 0, 0, 1, 1, 1, 1}	0	0000	
y5={0, 0, 1, 0, 0, 0, 0}	8	1000	
{0, 0, 1, 0, 0, 0, 1}	9	1001	
{0, 0, 1, 0, 0, 1, 0}	10	1010	
{0, 0, 1, 0, 0, 1, 1}	11	1011	
{0, 0, 1, 0, 1, 0, 0}	12	1100	
{0, 0, 1, 0, 1, 0, 1}	13	1101	
{0, 0, 1, 0, 1, 1, 0}	14	1110	
{0, 0, 1, 0, 1, 1, 1}	15	1111	
{0, 0, 1, 1, 0, 0, 0}	15	1111	
{0, 0, 1, 1, 0, 0, 1}	14	1110	
{0, 0, 1, 1, 0, 1, 0}	13	1101	
{0, 0, 1, 1, 0, 1, 1}	12	1100	
{0, 0, 1, 1, 1, 0, 0}	11	1011	
{0, 0, 1, 1, 1, 0, 1}	10	1010	
{0, 0, 1, 1, 1, 1, 0}	9	1001	
{0, 0, 1, 1, 1, 1, 1}	8	1000	
y6={0, 1, 0, 0, 0, 0, 0}	11	1011	
{0, 1, 0, 0, 0, 0, 1}	10	1010	
{0, 1, 0, 0, 0, 1, 0}	9	1001	
{0, 1, 0, 0, 0, 1, 1}	8	1000	
{0, 1, 0, 0, 1, 0, 0}	15	1111	
{0, 1, 0, 0, 1, 0, 1}	14	1110	
{0, 1, 0, 0, 1, 1, 0}	13	1101	
{0, 1, 0, 0, 1, 1, 1}	12	1100	
{0, 1, 0, 1, 0, 0, 0}	12	1100	

{0, 1, 0, 1, 0, 0, 1}	13	1101
{0, 1, 0, 1, 0, 1, 0}	14	1110
{0, 1, 0, 1, 0, 1, 1}	15	1111
{0, 1, 0, 1, 1, 0, 0}	8	1000
{0, 1, 0, 1, 1, 0, 1}	9	1001
{0, 1, 0, 1, 1, 1, 0}	10	1010
{0, 1, 0, 1, 1, 1, 1}	11	1011
{0, 1, 1, 0, 0, 0, 0}	3	0011
{0, 1, 1, 0, 0, 0, 1}	2	0010
{0, 1, 1, 0, 0, 1, 0}	1	0001
{0, 1, 1, 0, 0, 1, 1}	0	0000
{0, 1, 1, 0, 1, 0, 0}	7	0111
{0, 1, 1, 0, 1, 0, 1}	6	0110
{0, 1, 1, 0, 1, 1, 0}	5	0101
{0, 1, 1, 0, 1, 1, 1}	4	0100
{0, 1, 1, 1, 0, 0, 0}	4	0100
{0, 1, 1, 1, 0, 0, 1}	5	0101
{0, 1, 1, 1, 0, 1, 0}	6	0110
{0, 1, 1, 1, 0, 1, 1}	7	0111
{0, 1, 1, 1, 1, 0, 0}	0	0000
{0, 1, 1, 1, 1, 0, 1}	1	0001
{0, 1, 1, 1, 1, 1, 0}	2	0010
{0, 1, 1, 1, 1, 1, 1}	3	0011
y7={1, 0, 0, 0, 0, 0, 0}	13	1101
{1, 0, 0, 0, 0, 0, 1}	12	1100
{1, 0, 0, 0, 0, 1, 0}	15	1111
{1, 0, 0, 0, 0, 1, 1}	14	1110
{1, 0, 0, 0, 1, 0, 0}	9	1001
{1, 0, 0, 0, 1, 0, 1}	8	1000
{1, 0, 0, 0, 1, 1, 0}	11	1011
{1, 0, 0, 0, 1, 1, 1}	10	1010
{1, 0, 0, 1, 0, 0, 0}	10	1010
{1, 0, 0, 1, 0, 0, 1}	11	1011
{1, 0, 0, 1, 0, 1, 0}	8	1000
{1, 0, 0, 1, 0, 1, 1}	9	1001
{1, 0, 0, 1, 1, 0, 0}	14	1110
{1, 0, 0, 1, 1, 0, 1}	15	1111
{1, 0, 0, 1, 1, 1, 0}	12	1100
{1, 0, 0, 1, 1, 1, 1}	13	1101
{1, 0, 1, 0, 0, 0, 0}	5	0101
{1, 0, 1, 0, 0, 0, 1}	4	0100
{1, 0, 1, 0, 0, 1, 0}	7	0111
{1, 0, 1, 0, 0, 1, 1}	6	0110
{1, 0, 1, 0, 1, 0, 0}	1	0001
{1, 0, 1, 0, 1, 0, 1}	0	0000
{1, 0, 1, 0, 1, 1, 0}	3	0011
{1, 0, 1, 0, 1, 1, 1}	2	0010
{1, 0, 1, 1, 0, 0, 0}	2	0010
{1, 0, 1, 1, 0, 0, 1}	3	0011
{1, 0, 1, 1, 0, 1, 0}	0	0000
{1, 0, 1, 1, 0, 1, 1}	1	0001
{1, 0, 1, 1, 1, 0, 0}	6	0110
{1, 0, 1, 1, 1, 0, 1}	7	0111
{1, 0, 1, 1, 1, 1, 0}	4	0100

{1, 0, 1, 1, 1, 1, 1}	5	0101
{1, 1, 0, 0, 0, 0, 0}	6	0110
{1, 1, 0, 0, 0, 0, 1}	7	0111
{1, 1, 0, 0, 0, 1, 0}	4	0100
{1, 1, 0, 0, 0, 1, 1}	5	0101
{1, 1, 0, 0, 1, 0, 0}	2	0010
{1, 1, 0, 0, 1, 0, 1}	3	0011
{1, 1, 0, 0, 1, 1, 0}	0	0000
{1, 1, 0, 0, 1, 1, 1}	1	0001
{1, 1, 0, 1, 0, 0, 0}	1	0001
{1, 1, 0, 1, 0, 0, 1}	0	0000
{1, 1, 0, 1, 0, 1, 0}	3	0011
{1, 1, 0, 1, 0, 1, 1}	2	0010
{1, 1, 0, 1, 1, 0, 0}	5	0101
{1, 1, 0, 1, 1, 0, 1}	4	0100
{1, 1, 0, 1, 1, 1, 0}	7	0111
{1, 1, 0, 1, 1, 1, 1}	6	0100
{1, 1, 1, 0, 0, 0, 0}	14	1110
{1, 1, 1, 0, 0, 0, 1}	15	1111
{1, 1, 1, 0, 0, 1, 0}	12	1100
{1, 1, 1, 0, 0, 1, 1}	13	1101
{1, 1, 1, 0, 1, 0, 0}	10	1010
{1, 1, 1, 0, 1, 0, 1}	11	1011
{1, 1, 1, 0, 1, 1, 0}	8	1000
{1, 1, 1, 0, 1, 1, 1}	9	1001
{1, 1, 1, 1, 0, 0, 0}	9	1001
{1, 1, 1, 1, 0, 0, 1}	8	1000
{1, 1, 1, 1, 0, 1, 0}	11	1011
{1, 1, 1, 1, 0, 1, 1}	10	1010
{1, 1, 1, 1, 1, 0, 0}	13	1101
{1, 1, 1, 1, 1, 0, 1}	12	1100
{1, 1, 1, 1, 1, 1, 0}	15	1111
{1, 1, 1, 1, 1, 1, 1}	14	1110

**Appendix: Table 13. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 1$  and Distance  $d = 5$ ,  $t = (d-1)/2$ ,  $M = 2^k = 2$ , with ordered basis:**

$y_1$ =	0000001
$y_2$ =	0000010
$y_3$ =	0000100
$y_4$ =	0001000
$y_5$ =	0010000
$y_6$ =	0100000
$y_7$ =	1000000

{0, 0, 0, 0, 0, 0, 0}	0	•
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 0}	2	
{0, 0, 0, 0, 0, 1, 1}	3	
y3={0, 0, 0, 0, 1, 0, 0}	4	
{0, 0, 0, 0, 1, 0, 1}	5	
{0, 0, 0, 0, 1, 1, 0}	6	
{0, 0, 0, 0, 1, 1, 1}	7	
y4={0, 0, 0, 1, 0, 0, 0}	8	
{0, 0, 0, 1, 0, 0, 1}	9	
{0, 0, 0, 1, 0, 1, 0}	10	
{0, 0, 0, 1, 0, 1, 1}	11	
{0, 0, 0, 1, 1, 0, 0}	12	
{0, 0, 0, 1, 1, 0, 1}	13	
{0, 0, 0, 1, 1, 1, 0}	14	
{0, 0, 0, 1, 1, 1, 1}	15	
y5={0, 0, 1, 0, 0, 0, 0}	15	
{0, 0, 1, 0, 0, 0, 1}	14	
{0, 0, 1, 0, 0, 1, 0}	13	
{0, 0, 1, 0, 0, 1, 1}	12	
{0, 0, 1, 0, 1, 0, 0}	11	
{0, 0, 1, 0, 1, 0, 1}	10	
{0, 0, 1, 0, 1, 1, 0}	9	
{0, 0, 1, 0, 1, 1, 1}	8	
{0, 0, 1, 1, 0, 0, 0}	7	
{0, 0, 1, 1, 0, 0, 1}	6	
{0, 0, 1, 1, 0, 1, 0}	5	
{0, 0, 1, 1, 0, 1, 1}	4	
{0, 0, 1, 1, 1, 0, 0}	3	
{0, 0, 1, 1, 1, 0, 1}	2	
{0, 0, 1, 1, 1, 1, 0}	1	
{0, 0, 1, 1, 1, 1, 1}	0	•
y6={0, 1, 0, 0, 0, 0, 0}	16	
{0, 1, 0, 0, 0, 0, 1}	17	
{0, 1, 0, 0, 0, 1, 0}	18	
{0, 1, 0, 0, 0, 1, 1}	19	
{0, 1, 0, 0, 1, 0, 0}	20	
{0, 1, 0, 0, 1, 0, 1}	21	
{0, 1, 0, 0, 1, 1, 0}	22	
{0, 1, 0, 0, 1, 1, 1}	23	
{0, 1, 0, 1, 0, 0, 0}	24	
{0, 1, 0, 1, 0, 0, 1}	25	

{0, 1, 0, 1, 0, 1, 0}	26
{0, 1, 0, 1, 0, 1, 1}	27
{0, 1, 0, 1, 1, 0, 0}	28
{0, 1, 0, 1, 1, 0, 1}	29
{0, 1, 0, 1, 1, 1, 0}	30
{0, 1, 0, 1, 1, 1, 1}	31
{0, 1, 1, 0, 0, 0, 0}	31
{0, 1, 1, 0, 0, 0, 1}	30
{0, 1, 1, 0, 0, 1, 0}	29
{0, 1, 1, 0, 0, 1, 1}	28
{0, 1, 1, 0, 1, 0, 0}	27
{0, 1, 1, 0, 1, 0, 1}	26
{0, 1, 1, 0, 1, 1, 0}	25
{0, 1, 1, 0, 1, 1, 1}	24
{0, 1, 1, 1, 0, 0, 0}	23
{0, 1, 1, 1, 0, 0, 1}	22
{0, 1, 1, 1, 0, 1, 0}	21
{0, 1, 1, 1, 0, 1, 1}	20
{0, 1, 1, 1, 1, 0, 0}	19
{0, 1, 1, 1, 1, 0, 1}	18
{0, 1, 1, 1, 1, 1, 0}	17
{0, 1, 1, 1, 1, 1, 1}	16
y7={1, 0, 0, 0, 0, 0, 0}	32
{1, 0, 0, 0, 0, 0, 1}	33
{1, 0, 0, 0, 0, 1, 0}	34
{1, 0, 0, 0, 0, 1, 1}	35
{1, 0, 0, 0, 1, 0, 0}	36
{1, 0, 0, 0, 1, 0, 1}	37
{1, 0, 0, 0, 1, 1, 0}	38
{1, 0, 0, 0, 1, 1, 1}	39
{1, 0, 0, 1, 0, 0, 0}	40
{1, 0, 0, 1, 0, 0, 1}	41
{1, 0, 0, 1, 0, 1, 0}	42
{1, 0, 0, 1, 0, 1, 1}	43
{1, 0, 0, 1, 1, 0, 0}	44
{1, 0, 0, 1, 1, 0, 1}	45
{1, 0, 0, 1, 1, 1, 0}	46
{1, 0, 0, 1, 1, 1, 1}	47
{1, 0, 1, 0, 0, 0, 0}	47
{1, 0, 1, 0, 0, 0, 1}	46
{1, 0, 1, 0, 0, 1, 0}	45
{1, 0, 1, 0, 0, 1, 1}	45
{1, 0, 1, 0, 1, 0, 0}	44
{1, 0, 1, 0, 1, 0, 1}	43
{1, 0, 1, 0, 1, 1, 0}	42
{1, 0, 1, 0, 1, 1, 1}	41
{1, 0, 1, 0, 1, 1, 1}	40
{1, 0, 1, 1, 0, 0, 0}	39
{1, 0, 1, 1, 0, 0, 1}	38
{1, 0, 1, 1, 0, 1, 0}	37
{1, 0, 1, 1, 0, 1, 1}	36
{1, 0, 1, 1, 1, 0, 0}	35
{1, 0, 1, 1, 1, 0, 1}	34
{1, 0, 1, 1, 1, 1, 0}	33
{1, 0, 1, 1, 1, 1, 1}	32

{1, 1, 0, 0, 0, 0, 0}	48
{1, 1, 0, 0, 0, 0, 1}	49
{1, 1, 0, 0, 0, 1, 0}	50
{1, 1, 0, 0, 0, 1, 1}	51
{1, 1, 0, 0, 1, 0, 0}	52
{1, 1, 0, 0, 1, 0, 1}	53
{1, 1, 0, 0, 1, 1, 0}	54
{1, 1, 0, 0, 1, 1, 1}	55
{1, 1, 0, 1, 0, 0, 0}	56
{1, 1, 0, 1, 0, 0, 1}	57
{1, 1, 0, 1, 0, 1, 0}	58
{1, 1, 0, 1, 0, 1, 1}	59
{1, 1, 0, 1, 1, 0, 0}	60
{1, 1, 0, 1, 1, 0, 1}	61
{1, 1, 0, 1, 1, 1, 0}	62
{1, 1, 0, 1, 1, 1, 1}	63
{1, 1, 1, 0, 0, 0, 0}	63
{1, 1, 1, 0, 0, 0, 1}	62
{1, 1, 1, 0, 0, 1, 0}	61
{1, 1, 1, 0, 0, 1, 1}	60
{1, 1, 1, 0, 1, 0, 0}	59
{1, 1, 1, 0, 1, 0, 1}	58
{1, 1, 1, 0, 1, 1, 0}	57
{1, 1, 1, 0, 1, 1, 1}	56
{1, 1, 1, 1, 0, 0, 0}	55
{1, 1, 1, 1, 0, 0, 1}	54
{1, 1, 1, 1, 0, 1, 0}	53
{1, 1, 1, 1, 0, 1, 1}	52
{1, 1, 1, 1, 1, 0, 0}	51
{1, 1, 1, 1, 1, 0, 1}	50
{1, 1, 1, 1, 1, 1, 0}	49
{1, 1, 1, 1, 1, 1, 1}	48

**Appendix: Table 14. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 1$  and Distance  $d = 6$ ,  $t = (d-1)/2$ ,  $M = 2^k = 2$ , with ordered basis:**

$y_1$ =	0000001
$y_2$ =	0000010
$y_3$ =	0000100
$y_4$ =	0001000
$y_5$ =	0010000
$y_6$ =	0100000
$y_7$ =	1000000

{0, 0, 0, 0, 0, 0, 0}	0	.
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 0}	2	
{0, 0, 0, 0, 0, 1, 1}	3	
y3={0, 0, 0, 0, 1, 0, 0}	4	
{0, 0, 0, 0, 1, 0, 1}	5	
{0, 0, 0, 0, 1, 1, 0}	6	
{0, 0, 0, 0, 1, 1, 1}	7	
y4={0, 0, 0, 1, 0, 0, 0}	8	
{0, 0, 0, 1, 0, 0, 1}	9	
{0, 0, 0, 1, 0, 1, 0}	10	
{0, 0, 0, 1, 0, 1, 1}	11	
{0, 0, 0, 1, 1, 0, 0}	12	
{0, 0, 0, 1, 1, 0, 1}	13	
{0, 0, 0, 1, 1, 1, 0}	14	
{0, 0, 0, 1, 1, 1, 1}	15	
y5={0, 0, 1, 0, 0, 0, 0}	16	
{0, 0, 1, 0, 0, 0, 1}	17	
{0, 0, 1, 0, 0, 1, 0}	18	
{0, 0, 1, 0, 0, 1, 1}	19	
{0, 0, 1, 0, 1, 0, 0}	20	
{0, 0, 1, 0, 1, 0, 1}	21	
{0, 0, 1, 0, 1, 1, 0}	22	
{0, 0, 1, 0, 1, 1, 1}	23	
{0, 0, 1, 1, 0, 0, 0}	24	
{0, 0, 1, 1, 0, 0, 1}	25	
{0, 0, 1, 1, 0, 1, 0}	26	
{0, 0, 1, 1, 0, 1, 1}	27	
{0, 0, 1, 1, 1, 0, 0}	28	
{0, 0, 1, 1, 1, 0, 1}	29	
{0, 0, 1, 1, 1, 1, 0}	30	
{0, 0, 1, 1, 1, 1, 1}	31	
y6={0, 1, 0, 0, 0, 0, 0}	31	
{0, 1, 0, 0, 0, 0, 1}	30	
{0, 1, 0, 0, 0, 1, 0}	29	
{0, 1, 0, 0, 0, 1, 1}	28	
{0, 1, 0, 0, 1, 0, 0}	27	
{0, 1, 0, 0, 1, 0, 1}	26	
{0, 1, 0, 0, 1, 1, 0}	25	
{0, 1, 0, 0, 1, 1, 1}	24	
{0, 1, 0, 1, 0, 0, 0}	23	
{0, 1, 0, 1, 0, 0, 1}	22	

{0, 1, 0, 1, 0, 1, 0}	21
{0, 1, 0, 1, 0, 1, 1}	20
{0, 1, 0, 1, 1, 0, 0}	19
{0, 1, 0, 1, 1, 0, 1}	18
{0, 1, 0, 1, 1, 1, 0}	17
{0, 1, 0, 1, 1, 1, 1}	16
{0, 1, 1, 0, 0, 0, 0}	15
{0, 1, 1, 0, 0, 0, 1}	14
{0, 1, 1, 0, 0, 1, 0}	13
{0, 1, 1, 0, 0, 1, 1}	12
{0, 1, 1, 0, 1, 0, 0}	11
{0, 1, 1, 0, 1, 0, 1}	10
{0, 1, 1, 0, 1, 1, 0}	9
{0, 1, 1, 0, 1, 1, 1}	8
{0, 1, 1, 1, 0, 0, 0}	7
{0, 1, 1, 1, 0, 0, 1}	6
{0, 1, 1, 1, 0, 1, 0}	5
{0, 1, 1, 1, 0, 1, 1}	4
{0, 1, 1, 1, 1, 0, 0}	3
{0, 1, 1, 1, 1, 0, 1}	2
{0, 1, 1, 1, 1, 1, 0}	1
{0, 1, 1, 1, 1, 1, 1}	0
Y7={1, 0, 0, 0, 0, 0, 0}	32
{1, 0, 0, 0, 0, 0, 1}	33
{1, 0, 0, 0, 0, 1, 0}	34
{1, 0, 0, 0, 0, 1, 1}	35
{1, 0, 0, 0, 1, 0, 0}	36
{1, 0, 0, 0, 1, 0, 1}	37
{1, 0, 0, 0, 1, 1, 0}	38
{1, 0, 0, 0, 1, 1, 1}	39
{1, 0, 0, 1, 0, 0, 0}	40
{1, 0, 0, 1, 0, 0, 1}	41
{1, 0, 0, 1, 0, 1, 0}	42
{1, 0, 0, 1, 0, 1, 1}	43
{1, 0, 0, 1, 1, 0, 0}	44
{1, 0, 0, 1, 1, 0, 1}	45
{1, 0, 0, 1, 1, 1, 0}	46
{1, 0, 0, 1, 1, 1, 1}	47
{1, 0, 1, 0, 0, 0, 0}	48
{1, 0, 1, 0, 0, 0, 1}	49
{1, 0, 1, 0, 0, 1, 0}	50
{1, 0, 1, 0, 0, 1, 1}	51
{1, 0, 1, 0, 1, 0, 0}	52
{1, 0, 1, 0, 1, 0, 1}	53
{1, 0, 1, 0, 1, 1, 0}	54
{1, 0, 1, 0, 1, 1, 1}	55
{1, 0, 1, 1, 0, 0, 0}	56
{1, 0, 1, 1, 0, 0, 1}	57
{1, 0, 1, 1, 0, 1, 0}	58
{1, 0, 1, 1, 0, 1, 1}	59
{1, 0, 1, 1, 1, 0, 0}	60
{1, 0, 1, 1, 1, 0, 1}	61
{1, 0, 1, 1, 1, 1, 0}	62
{1, 0, 1, 1, 1, 1, 1}	63

{1, 1, 0, 0, 0, 0, 0}	63
{1, 1, 0, 0, 0, 0, 1}	62
{1, 1, 0, 0, 0, 1, 0}	61
{1, 1, 0, 0, 0, 1, 1}	60
{1, 1, 0, 0, 1, 0, 0}	59
{1, 1, 0, 0, 1, 0, 1}	58
{1, 1, 0, 0, 1, 1, 0}	57
{1, 1, 0, 0, 1, 1, 1}	56
{1, 1, 0, 1, 0, 0, 0}	55
{1, 1, 0, 1, 0, 0, 1}	54
{1, 1, 0, 1, 0, 1, 0}	53
{1, 1, 0, 1, 0, 1, 1}	52
{1, 1, 0, 1, 1, 0, 0}	51
{1, 1, 0, 1, 1, 0, 1}	50
{1, 1, 0, 1, 1, 1, 0}	49
{1, 1, 0, 1, 1, 1, 1}	48
{1, 1, 1, 0, 0, 0, 0}	47
{1, 1, 1, 0, 0, 0, 1}	46
{1, 1, 1, 0, 0, 1, 0}	45
{1, 1, 1, 0, 0, 1, 1}	44
{1, 1, 1, 0, 1, 0, 0}	43
{1, 1, 1, 0, 1, 0, 1}	42
{1, 1, 1, 0, 1, 1, 0}	41
{1, 1, 1, 0, 1, 1, 1}	40
{1, 1, 1, 1, 0, 0, 0}	39
{1, 1, 1, 1, 0, 0, 1}	38
{1, 1, 1, 1, 0, 1, 0}	37
{1, 1, 1, 1, 0, 1, 1}	36
{1, 1, 1, 1, 1, 0, 0}	35
{1, 1, 1, 1, 1, 0, 1}	34
{1, 1, 1, 1, 1, 1, 0}	33
{1, 1, 1, 1, 1, 1, 1}	32

**Appendix: Table 15. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 1$  and Distance  $d = 7$ ,  $t = (d-1)/2$ ,  $M = 2^k = 2$ , with ordered basis:**

$y_1$ =	0000001
$y_2$ =	0000010
$y_3$ =	0000100
$y_4$ =	0001000
$y_5$ =	0010000
$y_6$ =	0100000
$y_7$ =	1000000

{0, 0, 0, 0, 0, 0, 0}	0	.
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 0}	2	
{0, 0, 0, 0, 1, 1, 0}	3	
y3={0, 0, 0, 0, 1, 0, 0}	4	
{0, 0, 0, 0, 1, 0, 1}	5	
{0, 0, 0, 0, 1, 1, 0}	6	
{0, 0, 0, 0, 1, 1, 1}	7	
y4={0, 0, 0, 1, 0, 0, 0}	8	
{0, 0, 0, 1, 0, 0, 1}	9	
{0, 0, 0, 1, 0, 1, 0}	10	
{0, 0, 0, 1, 0, 1, 1}	11	
{0, 0, 0, 1, 1, 0, 0}	12	
{0, 0, 0, 1, 1, 0, 1}	13	
{0, 0, 0, 1, 1, 1, 0}	14	
{0, 0, 0, 1, 1, 1, 1}	15	
y5={0, 0, 1, 0, 0, 0, 0}	16	
{0, 0, 1, 0, 0, 0, 1}	17	
{0, 0, 1, 0, 0, 1, 0}	18	
{0, 0, 1, 0, 0, 1, 1}	19	
{0, 0, 1, 0, 1, 0, 0}	20	
{0, 0, 1, 0, 1, 0, 1}	21	
{0, 0, 1, 0, 1, 1, 0}	22	
{0, 0, 1, 0, 1, 1, 1}	23	
{0, 0, 1, 1, 0, 0, 0}	24	
{0, 0, 1, 1, 0, 0, 1}	25	
{0, 0, 1, 1, 0, 1, 0}	26	
{0, 0, 1, 1, 0, 1, 1}	27	
{0, 0, 1, 1, 1, 0, 0}	28	
{0, 0, 1, 1, 1, 0, 1}	29	
{0, 0, 1, 1, 1, 1, 0}	30	
{0, 0, 1, 1, 1, 1, 1}	31	
y6={0, 1, 0, 0, 0, 0, 0}	32	
{0, 1, 0, 0, 0, 0, 1}	33	
{0, 1, 0, 0, 0, 1, 0}	34	
{0, 1, 0, 0, 0, 1, 1}	35	
{0, 1, 0, 0, 1, 0, 0}	36	
{0, 1, 0, 0, 1, 0, 1}	37	
{0, 1, 0, 0, 1, 1, 0}	38	
{0, 1, 0, 0, 1, 1, 1}	39	
{0, 1, 0, 1, 0, 0, 0}	40	
{0, 1, 0, 1, 0, 0, 1}	41	

{0, 1, 0, 1, 0, 1, 0}	42
{0, 1, 0, 1, 0, 1, 1}	43
{0, 1, 0, 1, 1, 0, 0}	44
{0, 1, 0, 1, 1, 0, 1}	45
{0, 1, 0, 1, 1, 1, 0}	46
{0, 1, 0, 1, 1, 1, 1}	47
{0, 1, 1, 0, 0, 0, 0}	48
{0, 1, 1, 0, 0, 0, 1}	49
{0, 1, 1, 0, 0, 1, 0}	50
{0, 1, 1, 0, 0, 1, 1}	51
{0, 1, 1, 0, 1, 0, 0}	52
{0, 1, 1, 0, 1, 0, 1}	53
{0, 1, 1, 0, 1, 1, 0}	54
{0, 1, 1, 0, 1, 1, 1}	55
{0, 1, 1, 1, 0, 0, 0}	56
{0, 1, 1, 1, 0, 0, 1}	57
{0, 1, 1, 1, 0, 1, 0}	58
{0, 1, 1, 1, 0, 1, 1}	59
{0, 1, 1, 1, 1, 0, 0}	60
{0, 1, 1, 1, 1, 0, 1}	61
{0, 1, 1, 1, 1, 1, 0}	62
{0, 1, 1, 1, 1, 1, 1}	63
y7={1, 0, 0, 0, 0, 0, 0}	63
{1, 0, 0, 0, 0, 0, 1}	62
{1, 0, 0, 0, 0, 1, 0}	61
{1, 0, 0, 0, 0, 1, 1}	60
{1, 0, 0, 0, 1, 0, 0}	59
{1, 0, 0, 0, 1, 0, 1}	58
{1, 0, 0, 0, 1, 1, 0}	57
{1, 0, 0, 0, 1, 1, 1}	56
{1, 0, 0, 1, 0, 0, 0}	55
{1, 0, 0, 1, 0, 0, 1}	54
{1, 0, 0, 1, 0, 1, 0}	53
{1, 0, 0, 1, 0, 1, 1}	52
{1, 0, 0, 1, 1, 0, 0}	51
{1, 0, 0, 1, 1, 0, 1}	50
{1, 0, 0, 1, 1, 1, 0}	49
{1, 0, 0, 1, 1, 1, 1}	48
{1, 0, 1, 0, 0, 0, 0}	47
{1, 0, 1, 0, 0, 0, 1}	46
{1, 0, 1, 0, 0, 1, 0}	45
{1, 0, 1, 0, 0, 1, 1}	44
{1, 0, 1, 0, 1, 0, 0}	43
{1, 0, 1, 0, 1, 0, 1}	42
{1, 0, 1, 0, 1, 1, 0}	41
{1, 0, 1, 0, 1, 1, 1}	40
{1, 0, 1, 1, 0, 0, 0}	39
{1, 0, 1, 1, 0, 0, 1}	38
{1, 0, 1, 1, 0, 1, 0}	37
{1, 0, 1, 1, 0, 1, 1}	36
{1, 0, 1, 1, 1, 0, 0}	35
{1, 0, 1, 1, 1, 0, 1}	34
{1, 0, 1, 1, 1, 1, 0}	33
{1, 0, 1, 1, 1, 1, 1}	32

{1, 1, 0, 0, 0, 0, 0}	31
{1, 1, 0, 0, 0, 0, 1}	30
{1, 1, 0, 0, 0, 1, 0}	29
{1, 1, 0, 0, 0, 1, 1}	28
{1, 1, 0, 0, 1, 0, 0}	27
{1, 1, 0, 0, 1, 0, 1}	26
{1, 1, 0, 0, 1, 1, 0}	25
{1, 1, 0, 0, 1, 1, 1}	24
{1, 1, 0, 1, 0, 0, 0}	23
{1, 1, 0, 1, 0, 0, 1}	22
{1, 1, 0, 1, 0, 1, 0}	21
{1, 1, 0, 1, 0, 1, 1}	20
{1, 1, 0, 1, 1, 0, 0}	19
{1, 1, 0, 1, 1, 0, 1}	18
{1, 1, 0, 1, 1, 1, 0}	17
{1, 1, 0, 1, 1, 1, 1}	16
{1, 1, 1, 0, 0, 0, 0}	15
{1, 1, 1, 0, 0, 0, 1}	14
{1, 1, 1, 0, 0, 1, 0}	13
{1, 1, 1, 0, 0, 1, 1}	12
{1, 1, 1, 0, 1, 0, 0}	11
{1, 1, 1, 0, 1, 0, 1}	10
{1, 1, 1, 0, 1, 1, 0}	9
{1, 1, 1, 0, 1, 1, 1}	8
{1, 1, 1, 1, 0, 0, 0}	7
{1, 1, 1, 1, 0, 0, 1}	6
{1, 1, 1, 1, 0, 1, 0}	5
{1, 1, 1, 1, 0, 1, 1}	4
{1, 1, 1, 1, 1, 0, 0}	3
{1, 1, 1, 1, 1, 0, 1}	2
{1, 1, 1, 1, 1, 1, 0}	1
{1, 1, 1, 1, 1, 1, 1}	0

•

**Appendix: Table 16. Lexicographic-Greedy Code of Length  $n = 7$ ,  $k = 0$  and Distance  $d = 8$ ,  $t = (d-1)/2$ ,  $M = 2^k = 1$ , with ordered basis:**

$y_1$ =	0000001
$y_2$ =	0000010
$y_3$ =	0000100
$y_4$ =	0001000
$y_5$ =	0010000
$y_6$ =	0100000
$y_7$ =	1000000

{0, 0, 0, 0, 0, 0, 0}	0	.
y1={0, 0, 0, 0, 0, 0, 1}	1	
y2={0, 0, 0, 0, 0, 1, 0}	2	
{0, 0, 0, 0, 1, 1, 1}	3	
y3={0, 0, 0, 0, 1, 0, 0}	4	
{0, 0, 0, 0, 1, 0, 1}	5	
{0, 0, 0, 0, 1, 1, 0}	6	
{0, 0, 0, 0, 1, 1, 1}	7	
y4={0, 0, 0, 1, 0, 0, 0}	8	
{0, 0, 0, 1, 0, 0, 1}	9	
{0, 0, 0, 1, 0, 1, 0}	10	
{0, 0, 0, 1, 0, 1, 1}	11	
{0, 0, 0, 1, 1, 0, 0}	12	
{0, 0, 0, 1, 1, 0, 1}	13	
{0, 0, 0, 1, 1, 1, 0}	14	
{0, 0, 0, 1, 1, 1, 1}	15	
y5={0, 0, 1, 0, 0, 0, 0}	16	
{0, 0, 1, 0, 0, 0, 1}	17	
{0, 0, 1, 0, 0, 1, 0}	18	
{0, 0, 1, 0, 0, 1, 1}	19	
{0, 0, 1, 0, 1, 0, 0}	20	
{0, 0, 1, 0, 1, 0, 1}	21	
{0, 0, 1, 0, 1, 1, 0}	22	
{0, 0, 1, 0, 1, 1, 1}	23	
{0, 0, 1, 1, 0, 0, 0}	24	
{0, 0, 1, 1, 0, 0, 1}	25	
{0, 0, 1, 1, 0, 1, 0}	26	
{0, 0, 1, 1, 0, 1, 1}	27	
{0, 0, 1, 1, 1, 0, 0}	28	
{0, 0, 1, 1, 1, 0, 1}	29	
{0, 0, 1, 1, 1, 1, 0}	30	
{0, 0, 1, 1, 1, 1, 1}	31	
y6={0, 1, 0, 0, 0, 0, 0}	32	
{0, 1, 0, 0, 0, 0, 1}	33	
{0, 1, 0, 0, 0, 1, 0}	34	
{0, 1, 0, 0, 0, 1, 1}	35	
{0, 1, 0, 0, 1, 0, 0}	36	
{0, 1, 0, 0, 1, 0, 1}	37	
{0, 1, 0, 0, 1, 1, 0}	38	
{0, 1, 0, 0, 1, 1, 1}	39	
{0, 1, 0, 1, 0, 0, 0}	40	
{0, 1, 0, 1, 0, 0, 1}	41	

{0, 1, 0, 1, 0, 1, 0}	42
{0, 1, 0, 1, 0, 1, 1}	43
{0, 1, 0, 1, 1, 0, 0}	44
{0, 1, 0, 1, 1, 0, 1}	45
{0, 1, 0, 1, 1, 1, 0}	46
{0, 1, 0, 1, 1, 1, 1}	47
{0, 1, 1, 0, 0, 0, 0}	48
{0, 1, 1, 0, 0, 0, 1}	49
{0, 1, 1, 0, 0, 1, 0}	50
{0, 1, 1, 0, 0, 1, 1}	51
{0, 1, 1, 0, 1, 0, 0}	52
{0, 1, 1, 0, 1, 0, 1}	53
{0, 1, 1, 0, 1, 1, 0}	54
{0, 1, 1, 0, 1, 1, 1}	55
{0, 1, 1, 1, 0, 0, 0}	56
{0, 1, 1, 1, 0, 0, 1}	57
{0, 1, 1, 1, 0, 1, 0}	58
{0, 1, 1, 1, 0, 1, 1}	59
{0, 1, 1, 1, 1, 0, 0}	60
{0, 1, 1, 1, 1, 0, 1}	61
{0, 1, 1, 1, 1, 1, 0}	62
{0, 1, 1, 1, 1, 1, 1}	63
y7={1, 0, 0, 0, 0, 0, 0}	64
{1, 0, 0, 0, 0, 0, 1}	65
{1, 0, 0, 0, 0, 1, 0}	66
{1, 0, 0, 0, 0, 1, 1}	67
{1, 0, 0, 0, 1, 0, 0}	68
{1, 0, 0, 0, 1, 0, 1}	69
{1, 0, 0, 0, 1, 1, 0}	70
{1, 0, 0, 0, 1, 1, 1}	71
{1, 0, 0, 1, 0, 0, 0}	72
{1, 0, 0, 1, 0, 0, 1}	73
{1, 0, 0, 1, 0, 1, 0}	74
{1, 0, 0, 1, 0, 1, 1}	75
{1, 0, 0, 1, 1, 0, 0}	76
{1, 0, 0, 1, 1, 0, 1}	77
{1, 0, 0, 1, 1, 1, 0}	78
{1, 0, 0, 1, 1, 1, 1}	79
{1, 0, 1, 0, 0, 0, 0}	80
{1, 0, 1, 0, 0, 0, 1}	81
{1, 0, 1, 0, 0, 1, 0}	82
{1, 0, 1, 0, 0, 1, 1}	83
{1, 0, 1, 0, 1, 0, 0}	84
{1, 0, 1, 0, 1, 0, 1}	85
{1, 0, 1, 0, 1, 1, 0}	86
{1, 0, 1, 0, 1, 1, 1}	87
{1, 0, 1, 1, 0, 0, 0}	88
{1, 0, 1, 1, 0, 0, 1}	89
{1, 0, 1, 1, 0, 1, 0}	90
{1, 0, 1, 1, 0, 1, 1}	91
{1, 0, 1, 1, 1, 0, 0}	92
{1, 0, 1, 1, 1, 0, 1}	93
{1, 0, 1, 1, 1, 1, 0}	94
{1, 0, 1, 1, 1, 1, 1}	95

{1, 1, 0, 0, 0, 0, 0}	96
{1, 1, 0, 0, 0, 0, 1}	97
{1, 1, 0, 0, 0, 1, 0}	98
{1, 1, 0, 0, 0, 1, 1}	99
{1, 1, 0, 0, 1, 0, 0}	100
{1, 1, 0, 0, 1, 0, 1}	101
{1, 1, 0, 0, 1, 1, 0}	102
{1, 1, 0, 0, 1, 1, 1}	103
{1, 1, 0, 1, 0, 0, 0}	104
{1, 1, 0, 1, 0, 0, 1}	105
{1, 1, 0, 1, 0, 1, 0}	106
{1, 1, 0, 1, 0, 1, 1}	107
{1, 1, 0, 1, 1, 0, 0}	108
{1, 1, 0, 1, 1, 0, 1}	109
{1, 1, 0, 1, 1, 1, 0}	110
{1, 1, 0, 1, 1, 1, 1}	111
{1, 1, 1, 0, 0, 0, 0}	112
{1, 1, 1, 0, 0, 0, 1}	113
{1, 1, 1, 0, 0, 1, 0}	114
{1, 1, 1, 0, 0, 1, 1}	115
{1, 1, 1, 0, 1, 0, 0}	116
{1, 1, 1, 0, 1, 0, 1}	117
{1, 1, 1, 0, 1, 1, 0}	118
{1, 1, 1, 0, 1, 1, 1}	119
{1, 1, 1, 1, 0, 0, 0}	120
{1, 1, 1, 1, 0, 0, 1}	121
{1, 1, 1, 1, 0, 1, 0}	122
{1, 1, 1, 1, 0, 1, 1}	123
{1, 1, 1, 1, 1, 0, 0}	124
{1, 1, 1, 1, 1, 0, 1}	125
{1, 1, 1, 1, 1, 1, 0}	126
{1, 1, 1, 1, 1, 1, 1}	127

## Appendix: Table 17      Greedy Code Algorithm.

```

(* Definitions *)

p=Table[x=0,{x,1,50}];
q=Table[x=0,{x,1,50}];
g=Table[x=0,{x,1,50}];
ff=Table[x=0,{x,1,50},{y,1,5}];
f3f=Table[x=0,{x,1,50},{y,1,5}];
Maximumplus[x_] :=(Max[##]+1)&
[Table[g[[y]],{y,1,x-1}]];
testagain[x_,y_] :=(
q[[x]]=Abs[ggs[[y]]-f[[x]]];
p[[x]]=Apply[Plus,
Table[If[q[[x,i]]==1,1,0], {i,1,5}]];
If[p[[x]]<3,g[[x]]=Maximumplus[x],
g[[x]]=g[[x]]]);
testgreedy[x_] :=
(Print["Here in testgreedy"];
p=Table[i=0,{i,1,50}];
q=Table[i=0,{i,1,50}];Print["ff's=",
ff[[1]],ff[[2]]];
greed=
Table[q[[y]]=Abs[ff[[y]]-f[[x]]];
p[[y]]=Apply[Plus,
Table[If[q[[y,i]]==1,1,0], {i,1,5}],
{y,y1,y2}]];
y3=Length[greed];
mm=Table[If[greed[[i]]<3,1,0],{i,1,y3}];
nn=Apply[Plus,Table[If[mm[[i]]>=1,1,0],
{i,1,y3}]];
If[nn>0,g[[x]]=Maximumplus[x],g[[x]]=yy]);
testzero[x_] :=(q[[x]]=
Abs[f[[1]]-f[[x]]];p[[x]]=Apply[Plus,
Table[If[q[[x,i]]==1,1,0], {i,1,5}]];
g[[x]]=p[[x]];
If[p[[x]]<3,g[[x]]=Maximumplus[x],g[[x]]=0]);
testmax[x_,y_] :=(q[[x]]=Abs[f[[y]]-f[[x]]];
p[[x]]=Apply[Plus,
Table[If[q[[x,i]]==1,1,0], {i,1,5}]];
If[p[[x]]<3,g[[x]]=Maximumplus[x],
g[[x]]=g[[y]]]);
RepeatTest[x_] :=(tempg=
Table[testmax[x,y], {y,1,x-1}]);
testall[x_] :=
Do[f3f[[i]]=f[[posx[[i,1]]]],{i,1,y5}];
p=Table[i=0,{i,1,50}];
q=Table[i=0,{i,1,50}];
greed=
Table[q[[y]]=Abs[f3f[[y]]-f[[x]]];
p[[y]]=Apply[Plus,
Table[If[q[[y,i]]==1,1,0], {i,1,5}]],

```

```

    {y,y1,y5}];
y3=Length[greed];
mm=Table[If[greed[[i]]<3,1,0],{i,1,y3}];
nn=Apply[Plus,Table[If[mm[[i]]>=1,1,0],
{i,1,y3}]];
If[nn>0,g[[x]]=Maximumplus[x],g[[x]]=g[[x]]]
);

(* Program *)

bb=32;
(testzero[bb]; RepeatTest[bb]; x1=Length[tempg]; i=1;
yy=Min[tempg];
gtrunc=Take[g,bb-1];
posit=Position[gtrunc,yy]; howmany=Length[posit];
y1=1; y2=howmany;
ff=Table[x=0,{x,1,50},{y,1,5}];
Table[ff[[x]]=f[[posit[[x,1]]]],{x,1,y2}];
xtempx=tempg;
Label[again];
yy=Min[xtempx];
posit=Position[g,yy]; howmany=Length[posit];
y1=1; y2=howmany;
Table[ff[[x]]=f[[posit[[x,1]]]],{x,1,y2}];
posit=Position[xtempx,yy]; howmany=Length[posit];
y1=1; y2=howmany;
testgreedy[bb];
gtrunc=Take[g,bb-1];
posx=Position[gtrunc,g[[bb]]];
y5=Length[posx]; Print["y5=",y5];
If[y5<=1,g[[bb]]=g[[bb]],testall[bb]];
xtempx=Delete[xtempx,posit]; i=i+1;
If[g[[bb]]==Maximumplus[bb] && i<x1 &&
Length[xtempx]>0,
Goto[again]]; g[[bb]]

```